"AS-BUILT" DESIGN SPECIFICATION OF THE CAMS/CAS INTERFACE TAPE REPORT GENERATION PROGRAM FOR LACIE 8

Job Order 71-593

(TIRFs 78-0016 & 78-0017)

(This document supersedes LEC-12022)

Prepared By

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION

SCIENCE AND APPLICATIONS DIRECTORATE

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

May 1978 LEC-12267

JSC-14238

"AS-BUILT" DESIGN SPECIFICATION OF THE CAMS/CAS INTERFACE TAPE REPORT GENERATION PROGRAM FOR LACIE 8

Job Order 71-593

(TIRFs 78-0016 & 78-0017)

PREPARED BY

K. P. Eckel

APPROVED BY

for James A Wilkinson

Philip L. Krumm, Acting Supervisor
Scientific Applications Section

Prepared By
Lockheed Electronics Company, Inc.
For
Earth Observations Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

May 1978 LEC-12267

CONTENTS

Section Page

Section Page

Section Page

1. SCOPE

1.1 GENERAL

This document is the "as-built" design specification of the CAMS/CAS Interface Tape Report Generation Program for LACIE 7.

2. APPLICABLE DOCUMENTS

- TIRF 77-0052
- TIRF 77-0040
- Specification for the CAMS/CAS Interface Tape Report Generation Program - LEC-9151
- CAMS/CAS Interface Control Tape format specification in Earth Resources Data Fromat Control Book Volume 1 (PHO-TR543, Rec. A. Change 3)
- TIRF 78-0010
- CAMS/CAS Interface Tape Printout after LACIE 6A JSC Memorandum, SF4-77-7-13, 7/21/77.
- "As-Built" Design Specification of the CAMS/CAS interface tape Report Generation Program LEC-11292
- Acceptance Test Specification For CAMS/CAS Interface tape report generation Program for LACIE 7-LEC-11787
- Clafification and Prioritization of LACIE 7 CCIT Report JSC memorandum, SF4-77-11-8, 11/4/77.
- TIRF 78-0016, Intermediate Values of Discriminate Function, Feb 78.
- TIRF 78-0017, CAMRPT Subclass Category Expansion, Feb 78.
- Acceptance Test Specification for subclass category expansion and Intermediate value display values to CAMS/CAS Interface Tape Report Program. LEC-12164.
- TIRF 78-0026, Combined Categories
- Acceptance Test Specification for the Combined Categories change to the CAMS/CAS Interface Tape Report Program. LEC-12613

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

N/A

3.2 SOFTWARE DESCRIPTION

The purpose of this program is to produce CAMS reports from data on the CAMS/CAS interface tape.

3.2.1 SOFTWARE COMPONENT NO. 1 (CAMRPT)

The main program CAMRPT reads control cards, locates segment data on the input tape and calls subroutines to generate requested reports.

3.2.1.1 Linkages

CAMRPT calls subroutines CDRED, CAMREC, BAUEXT, CLURES, STDATA, CONRED, and DOTRAY.

3.2.1.2 Interfaces

N/A

3.2.1.3 Inputs

CAMRPT control cards are: SEGMENT XXXX, RECORD ID XXXXXX XXXXXX, ALL, END. CAMS/CAS interface tape records are inputs to CAMRPT. See reference 4 in section 2, for record formats.

3.2.1.4 Outputs

An error message is output indicating a bad data card. If a requested segment is not on the input tape, the program writes a message to that effect.

3.2.1.5 Storage Requirements

Total space allocated is 8474 bytes.

3.2.1.6 Description

CAMRPT is the CAMRPT main program. The program sets the printout option indicator PRTOUT to 0 initially. In this mode the output of some reports is conditional. On the first call to tape read, subroutine CDRED, the program reads two data cards specifying the device code (M or X) and the unit number (0 to 1), then calls CONRED to read constants to be used in the calculations for DW & DS and the subclass categories to be combined. CAMPRT next reads a program control card and tests the first non-blank character for one of the following: S, R, A, or E. If the card is blank or the first character is not one of the above, the program prints an error message on the line printer and stops.

The action taken for each control card is given below. Note that if a control card other than E is read in, the printout option indicator PRTOUT is set to 1. This is the option to output all reports, including conditional reports.

- S - Option indicator PRTOUT is set to 1. The program obtains the segment number from the input card. The program searches the input tape for a recognition segment record whose segment number matches the control card segment number. If a match is not found, a message is printed and the program goes to read the next control card. If a match is found, the reports for the segment requested are generated. The program then reads the next control card.

  program goes to read the next control card. If a match is found, the reports for the segment requested are generated. The program then reads the next control card.

- R - The action taken is the same as for the S card above, except that the record identification number is used instead of the segment number.

- A - Option indicator PRTOUT is set to 1. Beginning with the segment on the tape at which the tape is currently positioned, the program generates reports for that

segment and all the following segments. When the second tape end of file, indicating end of data, is reached, the program rewinds the tape and returns to read the next control card.

- E - If the printout option indicator PRTOUT is 0, the program generates reports in the limited printout mode, rewinds the tape, and then stops. If PRTOUT is 1, the tape is rewound and the program stops.

The tape read subroutine CDRED tests all records for valid characters. If any invalid data is encountered an error message is output to the line printer to inform the user that the data for that segment or record is questionable. Any invalid characters in the record are converted to ones (1) and normal processing is resumed.

To generate reports for a segment the program first calls CAMREC to process recognition segment records. The processing entails generation of the optional classification Data report, the standard Separability Report and the standard header sheet for the report.

After CAMREC, the programming calls BAUEXT to process the Bias correction results records. Data from the Bias correction results records is saved in common blocks BIAS and Dummy. CAMRPT next calls CLURES to process the clustering results records and to generate the conditional cluster report. In addition CLURES saves cluster dot data in common block CLDOT for later use in the optional Dot report. Next STDATA is called to process the statistics records and generate the optional Statistics Report. Finally DOTRAY is called to process the Dot subset records and generate the standard Dot Label/Classification, Bias Correction Classification, Dot Label/Cluster and Bias Correction Cluster Reports.

In the limited printout mode the decision as to whether or not to output the optional reports is not made until the percentage of

correctly classified Dots (PCC-1 & PCC-2 for Bias Correction Classification report) or percentage of correctly clustered Dots (PCC-1 & PCC-2 for Bias Correction Cluster Report) are calculated in subroutine DOTRAY. If any of the values are less than 80%, PRTOUT is set to 2 in DOTRAY and the conditional reports are read from disc and written to the line printer. In the full printout mode (PRTOUT=1) the conditional reports are always retrieved from disc and printed out. If PRTOUT was = 2 it is reset to 0 after the report for a segment has been output.

3.2.1.7 Flowcharts

See Flow Diagram 1.

3.2.1.8 Listing

See Appendix A.

3.2.2 SOFTWARE COMPONENT NO. 2 (CAMREC)

This program processess classification results contained in recognition segment records, outputs the conditional Classification Data report, saves data from the recognition segment records for use in generating the standard Separability Report and the standard CAMS Interface Report Header sheet.

3.2.2.1 Linkages

CAMREC is called by CAMPRT and calls subroutines BIAPRT, CPIPO, MV, CDRED, BNT and SEPRPT.

3.2.2.2 Interfaces

N/A

3.2.2.3 Inputs

Recognition segment records, containing subclass a priori and threshold values, subclass related classification results, and feature selection Bhattacharyya separability data for available acquisitions.

3.2.2.4 Outputs

CAMS Interface Report Header and a Classification Data report.

3.2.2.5 Storage Requirements

Total space allocated is 7397 bytes.

3.2.2.6 Description

CAMREC is called with the first recognition segment record for the segment to be processed residing in array IBUF. CAMREC first calls BIAPRT with PASS=1 to have the report heading, segment

number, record ID, and acquisition dates output. Next CAMREC saves the number of channels used in classification and the Bhattacharyya separability data from the first recognition segment record for later use in generation of the Separability Report.

Title and column headings for the classification section of the report are written out by CAMREC. Processing of classification results begins by setting the location in array IBUF of the first subfield containing subclass related results. Subfield contents are accessed by calling CPIPO. CPIPO returns the class portion of the subclass name and the counts PI and PO of pixels classified into, and thresholded out of the subclass. If the first character of the class name is X, PI is added to the X category pixel count. If the category is W, for wheat, then the count for the first wheat class is set to PI and the wheat class name is saved in CLIST. PO is added to the total of pixels threshold, TC, in the COMMON blocks CBIAS.

In processing for the second, and subsequent subclasses, the program calls CPIPO to get the next class name, checks to see if it is wheat, and, if so, compares it to the last class name in CLIST. If it is not the same, the new name is saved in CLIST and the class index is incremented by 1. This causes wheat class pixel count PI to be tallied in the next results array location.

After all classification data has been processed the feature selection Bhattacharyya separability data is saved from the last recognition segment record and SEPRPT is called to generate the normal Separability Report.

3.2.2.7 Flowcharts

N/A

3.2.2.8 Listing

See Appendix A.

3.2.3 SOFTWARE COMPONENT NO. 3 (BAUEXT)

This program saves data contained in the clustering bias correction and classification bias correction results records.

3.2.3.1 Linkages

BAUEXT is called by CAMRPT. It calls CDRED.

3.2.3.2 Interfaces

N/A

3.2.3.3 Inputs

Clustering Bias correction and Classification Bias Correction results records.

3.2.3.4 Outputs

None.

3.2.3.5 Storage Requirements

Total space allocated is 6861 bytes.

3.2.3.6 Description

BAUEXT is called when the main program reads the first clustering Bias correction result record. BAUEXT saves the following data from both the Clustering Bias Correction and Classification Bias Correction results records for up to 8 categories of interest plus the "designated other" and "unclassified" category"

Pixel Population
Bias corrected estimator
Machine estimate
Random estimate
Variable of bias corrected estimate
Variance

In addition the number of categories of interest and the character used for the categories of interest are saved. All data is saved in common blocks CBIAS arrays. This data is used by CALC for certain calculations and by BIAPRT for output of the normal Bias Correction reports.

3.2.3.7 Flowcharts

N/A

3.2.3.8 Listing

See Appendix A.

3.2.4 SOFTWARE COMPONENT NO. 4 (CLURES)

This program processes the cluster results records and generates the conditional cluster report.

3.2.4.1 Linkages

CLURES is called by CAMRPT. It calls CDRED.

3.2.4.2 Interfaces

N/A

3.2.4.3 Inputs

Cluster results records.

3.2.4.4 Outputs

The conditional Cluster report.

3.2.4.5 Storage Requirements

Total space allocated is 10966 bytes.

3.2.4.6 Description

CLURES is called when the main routine reads the first cluster results record. The program decodes ALSETS, the total number of clusters, and SETSR, the number of clusters in the current record. The routine then outputs the cluster report header, ALSETS as clusters generated and any options used. Next CLURES saves all data for each cluster for later output. When all clusters in the current record have been processed, another cluster results record is read in and processed as above.

After all cluster have been processed and if the cluster/dot report option is set, four additional cluster results records need to be processed. The processing consists of saving all dot

information in an array called DOTBUF for later output. In addition each dots cluster assignment is transfered to the common blocks CLCOM for later use in a different report. Finally when all cluster results records are processed in the above manner the cluster information is output as follows. For each cluster the program outputs the cluster name, the Labeling dot match name, L1 distance, categorie used, brightness and greenness numbers for all Acquisitions used and information on all dots in the cluster. The clustering channel list is written at the end of the report.

3.2.4.7 Flowchart

See Flow Diagram 2.

3.2.4.8 Listing

See Appendix A.

3.2.5 SOFTWARE COMPONENT NO. 5 (STDATA)

This subroutine formats and outputs field and subclass statistics data.

3.2.5.1 Linkages

STDATA is called by CAMRPT. STDATA calls subroutines KNT, MDTTL, MEAN, POP, CDRED, SNAME, FANME, STDMP, and BNT.

3.2.5.2 Interfaces

N/A

3.2.5.3 Inputs

The statistics record, containing, for fields or for subclasses, the population and values of the mean and standard deviation by channel.

3.2.5.4 Outputs

The conditional statistics report.

3.2.5.5 Storage Requirements

Space allocated is 8360 bytes.

3.2.5.6 Description

STDATA is called from CAMRPT. By means of decode statements, the program converts several variables from input character format in IBUF to integers. The variables are ALSETS, the total number of statistics sets, SETSR, the number of sets in the current record, and NCH, the number of channels. STDATA calls 5 subroutines to move data from input record subfields to print buffers. SNAME and FNAME move name data and insert SUBCL and FIELD designations in the print buffer. POP is called to move

population data. MDTTL is called to supply column headings for means and standard deviations, which are transferred to a print buffer by MEAN. MEAN also puts decimal points where needed. The variable DSETS, set to 5, controls the number of statistics sets to be accumulated before outputting the print buffers. When the current record statistics sets counter reaches SETSR, and ALSETS sets have not yet been processed, STDATA calls CDRED to read the next statiscics record from tape. In addition, the pixel population and classified percentage for the category "unassigned" in the bias correction cluster report is calculated and saved in the common blocks CBIAS.

3.2.5.7 Flowcharts

N/A

3.2.5.8 Listing

See Appendix A.

3.2.6 SOFTWARE COMPONENT NUMBER 6 (DOTRAY)

This program processes Dot Data records and generates the conditional Dot Report.

3.2.6.1 Linkage

DOTRAY is called by CAMRPT. It calls BIAPRT, CALC and CDRED.

3.2.6.2 Interfaces

Dot Data records.

3.2.6.3 Inputs

N/A

3.2.6.4 Outputs

The Conditional Dot report.

3.2.6.5 Storage Requirements

Total space allocated is 6986 Bytes.

3.2.6.6 Description

DOTRAY is called by CAMPRT after reading the first Dot Data record. DOTRAY next outputs the report header to the top of the next page and processess the Dot Data records until all 209 dots are processed. DOTRAY also saves the dot Lable, dot type and classification for each dot in the common block CBIAS for later use by BIAPRT & CALC.

The processing involves outputting the following for each of 209 dots:

Dot number, line and pixel number for the dot, type and label (if any) for the dot, cluster and classification as well as the greenness and brightness of up to 4 acquisition for the Dot. Each record contains data for 15 dots. After the 15 dots in the record has been processed and output the next record is read in via CDRED and processed. After all dots are processed DOTRAY

calls CALC to perform calculation for the bias correction reports then calls BIAPRT to output the bias correction reports. Finally, DOTRAY check the value PRTOUT. If PRTOUT is 1 DOTRAY returns to the main program. If PRTOUT is not 1 all valid PCC values (percentages of TYPE 1 & TYPE 2 correctly classified or clustered dots) are tested. If any of the valid PCC's are less than 80%, then PRTOUT is set to 2, to indicate to the main program that the conditional reports are to be read from the disk and output to the line printer.

3.2.6.7 Flowcharts

N/A

3.2.6.8 Listing

See Appendix A

3.2.7 SOFTWARE COMPONENT 7 (SEPRPT)

This program processess the separability data which was saved for it by CAMREC and outputs the separability report.

3.2.7.1 Linkage

SEPRPT is called by CAMREC.

3.2.7.2 Interfaces

N/A

3.2.7.3 Inputs

Segment Recognition record number 1.

3.2.7.4 Outputs

The normal Separability report.

3.2.7.5 Storage Requirements

Total space allocated in 588 bytes.

3.2.7.6 Description

SEPRPT is called by CAMREC after all separability data has been saved. SEPRPT then outputs the separability report which consists of channel combinations for up 4 (16 channels) and selection Battacharyya separability data.

3.2.7.7 Flowcharts

N/A

3.2.7.8 Listings

See Appendix A.

3.2.8 SOFTWARE COMPONENT NO. 8 (CALC)

This subroutine calculates data necessary for the output of the Bias Correction Classification and Bias Correction Cluster reports.

3.2.8.1 Linkage

CALC is called by DOTRAY.

3.2.8.2 Interface

N/A

3.2.8.3 Inputs

Dot Data from common blocks CBIAS.

3.2.8.4 Output

None

3.2.8.5 Storage

Total space allocated is 9248 bytes.

3.2.8.6 Description

CALC performs 2 identical calculations on different sets of data. The first data set involves Classification data and the second involves Cluster data. CALC first sets all needed variables to zero. If the flag CLADUM is equal to 1, no classification calculations are performed and CALC goes directly to the cluster calculations. Otherwise CALC checks "category of interest" for the presence of an "N", or a match of a member of the subclass categories to be combined, and sets pointers and indication as necessary. Next certain arrays are zeroed.

Following are the variables calculated for each of 209 dots. It is understood, in every case, that the conditions apply to DOTS which

are labeled, that is, LBLED (I) is not blank, and to DOTS not classified as DU or DO.

- NTYP1 - The number of DOTS which are either type 1 or type 3.
- NTYP2 - The number of type 2 DOTS
- NAII - The number of type 1 and type 3 DOTS whose label and classification are the same.
- NGIJ - The number of type 1 and type 3 "GRAIN TYPE" DOTS whose label and classification are not the same.
- NOCL - The number of type 2 DOTS which are both labeled and classified.

In addition when calculating the above for all dots a bias correction vector table is calculated. This table consists of a two dimensional array and contains summations of dots that have valid labels as the first index and valid classification as the second index.

In addition the following variables are calculated: It is understood that classified does not mean "threshold."

- ALGT - The number of TYPE 2 dots labeled in a cateogry to be combined and classified in a category to be combined.
- ALBG - The number of TYPE 2 dots labeled with any "category used" and classified in a category to be combined.
- ALNT - The number of TYPE 2 dots labeled in any "category used" other than a category to be combined and classified in any "category used" other than a category to be combined.
- ALNB - The number TYPE 2 dots labeled in any "category used" and classified in any "category used" other than a category to be combined.

The subroutine computes the corrected percentages, uncorrected populations, variances, uncorrected percentages and random sample data for all "categories used" plus "combined category."

CALC now performs the identical calculations on the cluster data after checking CLUDUM as outlined above.

3.2.8.7 Flowchart

See Flow diagram No. 3.

3.2.8.8 Listing

See Appendix A.

3.2.9 SOFTWARE COMPONENT NO. 9 (BIAPRT)

This program outputs the report header sheet, Label/Classification table, Lable/Cluster table and the Bias Correction reports.

3.2.9.1 Linkage

BIAPRT is called by CAMREC & DOTRAY. It calls BNT.

3.2.9.2 Interface

N/A

3.2.9.3 Inputs

Segment recognition records and common blocks CBIAS, Dummy and CLCOM.

3.2.9.4 Outputs

Report header sheet, TYPE1 and TYPE2 Dot Label/Classification report, Bias Correction Classification Report, TYPE1 and TYPE2 Dot Label/Cluster report and the Bias Correction Cluster reports.

3.2.9.5 Storage Requirements

Total space allocated in 9965 bytes.

3.2.9.6 Description

When CAMREC calls BIAPRT it sets PASS=1. This causes BIAPRT to output the report header which contains tape number, DPAR No., record ID, segment number and all acquisition dates. This information is retrieved from the segment recognition record which had been read into IBUF.

When DOTRAY calls BIAPRT it sets PASS=2. In this mode of operation up to 2 similar reports can be generated. If CLADUM is equal to 1 no classification report is generated.

Otherwise DOTRAY generates an 11 by 19 matrix of user label/classification entries for type 1 and type 3 DOTS. A similar Matrix is generated for type 2 DOTS which also includes type 0 DOTS. DOTS with a classification label of DU or DO do not appear in either matrix.

Next the bias correction classification report is output using data calculated by CALC and stored in common blocks CBIAS. The report consists entries for all "categories of interest", DO, TH, UN and 'combined' as follows:

Pixel population, classified and corrected percentages, variance and random sample estimate.

Also the alpha value matrix is output, and PCC values, DW and DS and lastly the Bias correction vectors and totals.

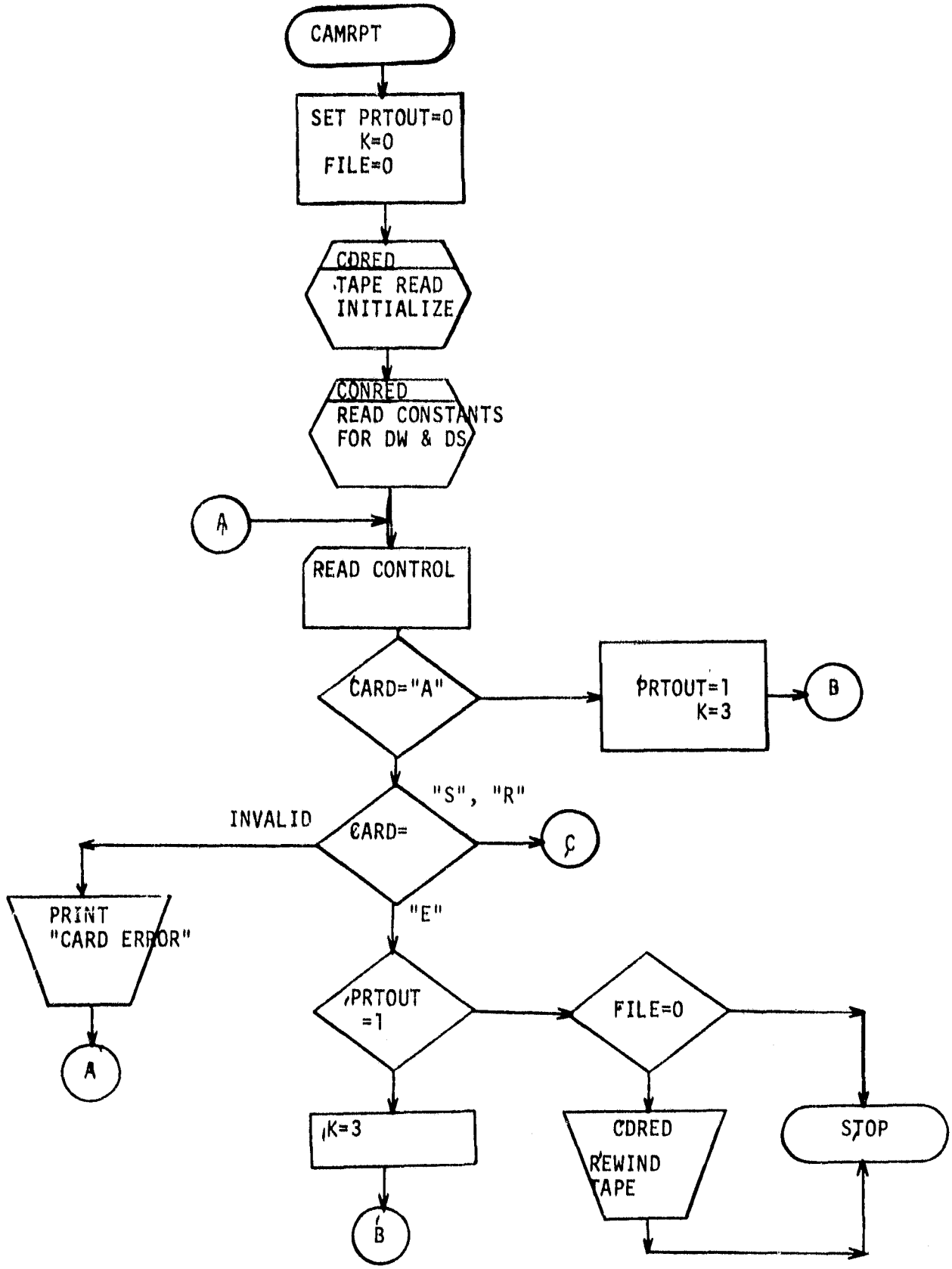
A similar report is generated for the cluster data if CLUDUM is not equal to 1.

3.2.9.7 Flowcharts

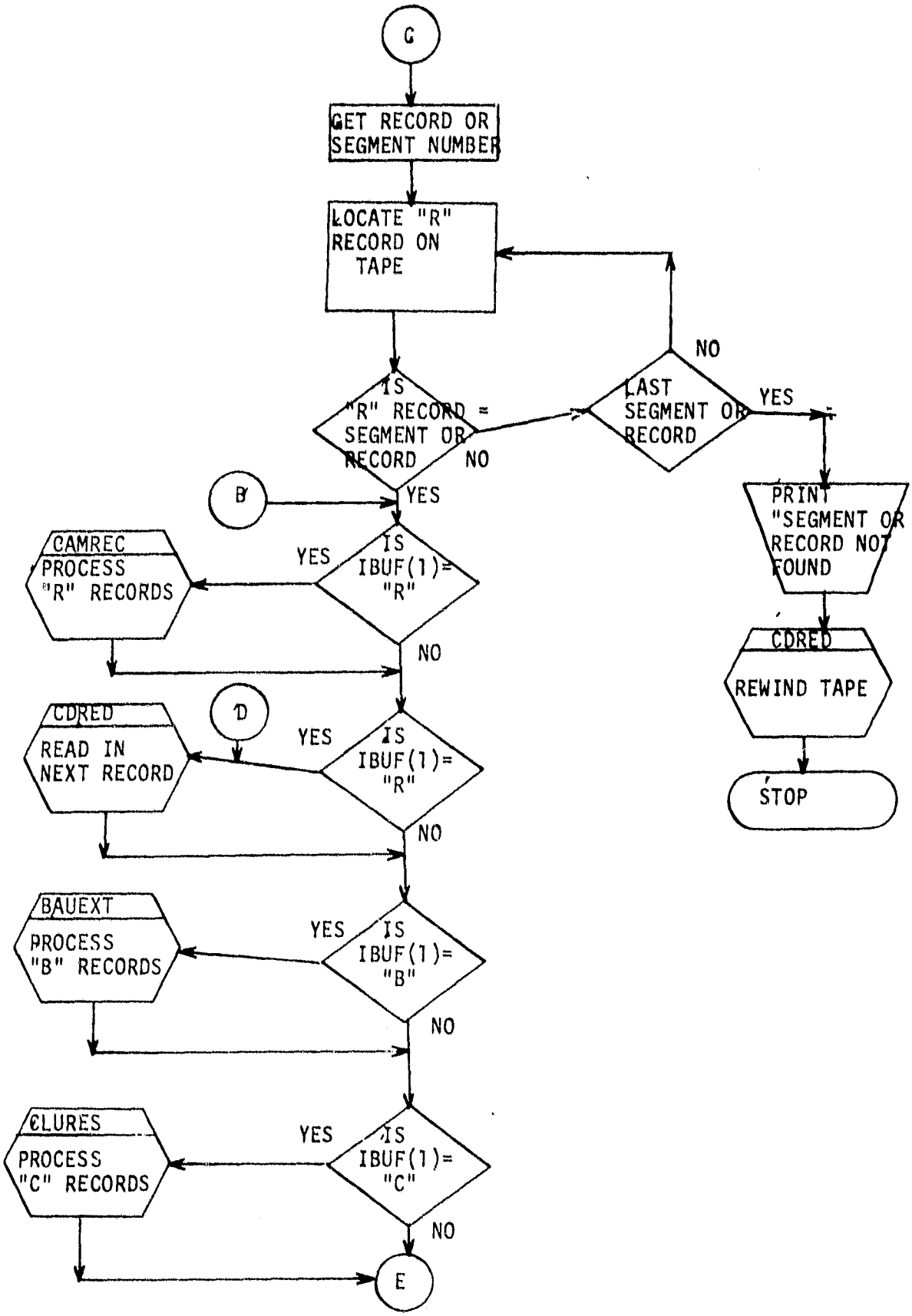
See flow diagram 4.

3.2.9.8 Listing

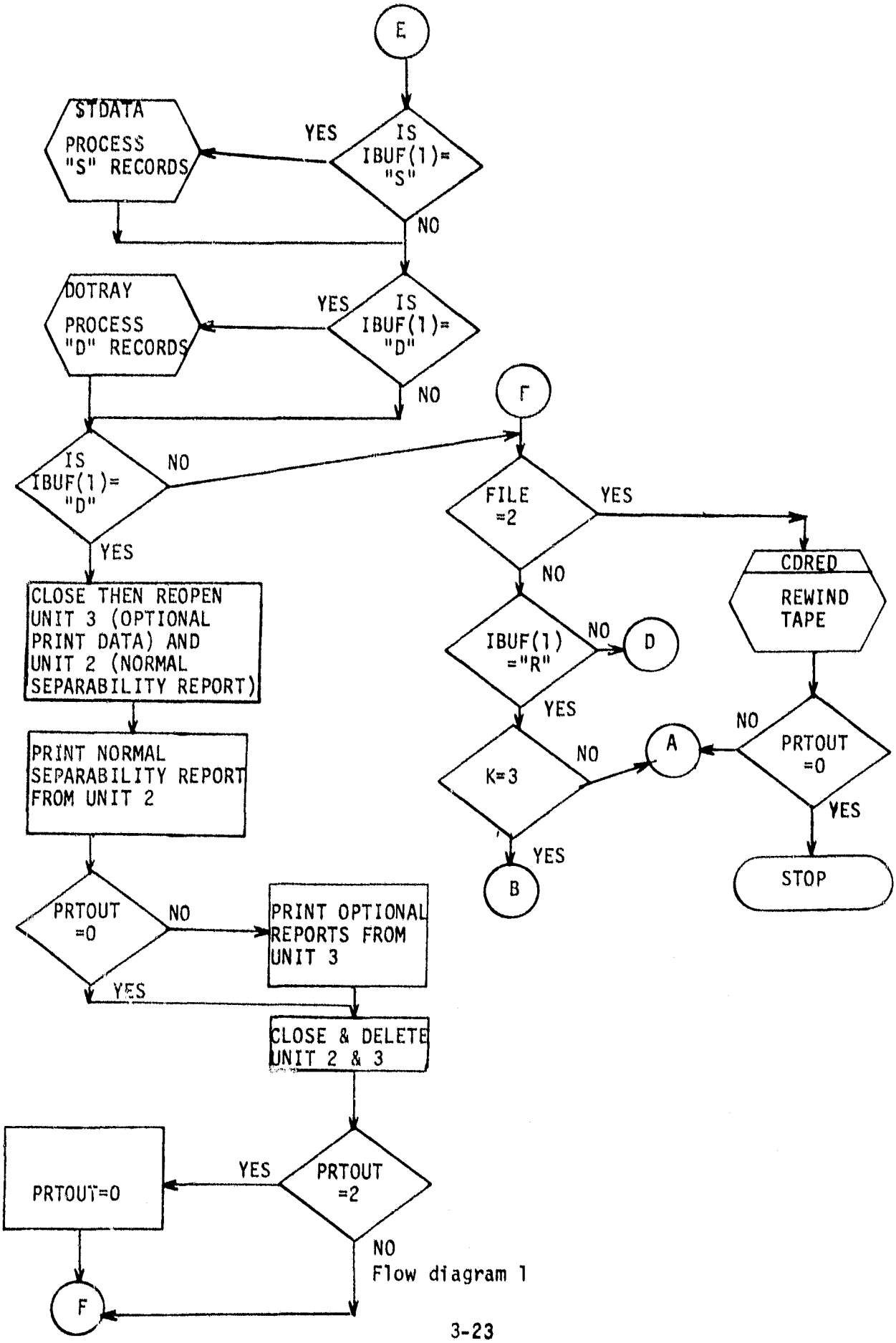
See Appendix A.



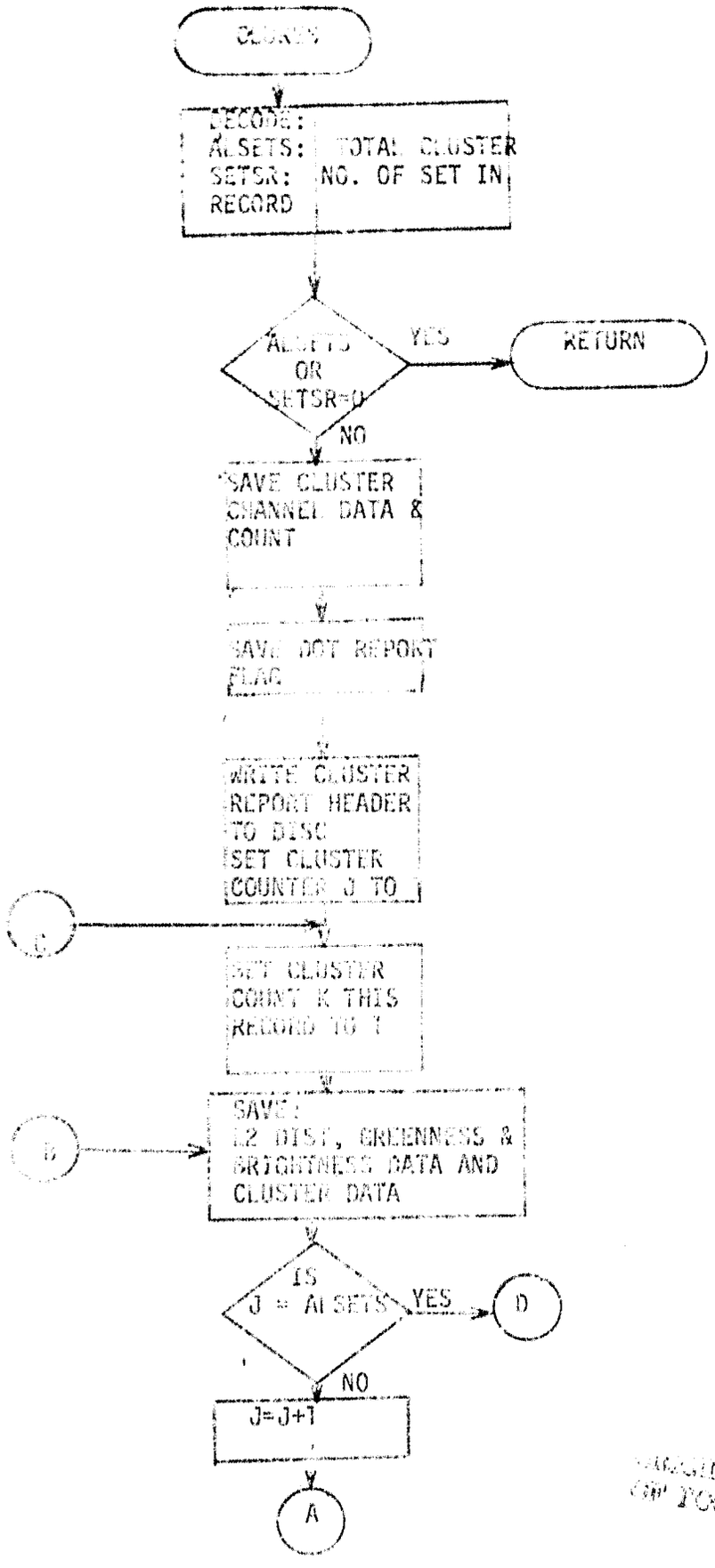
Flow diagram 1



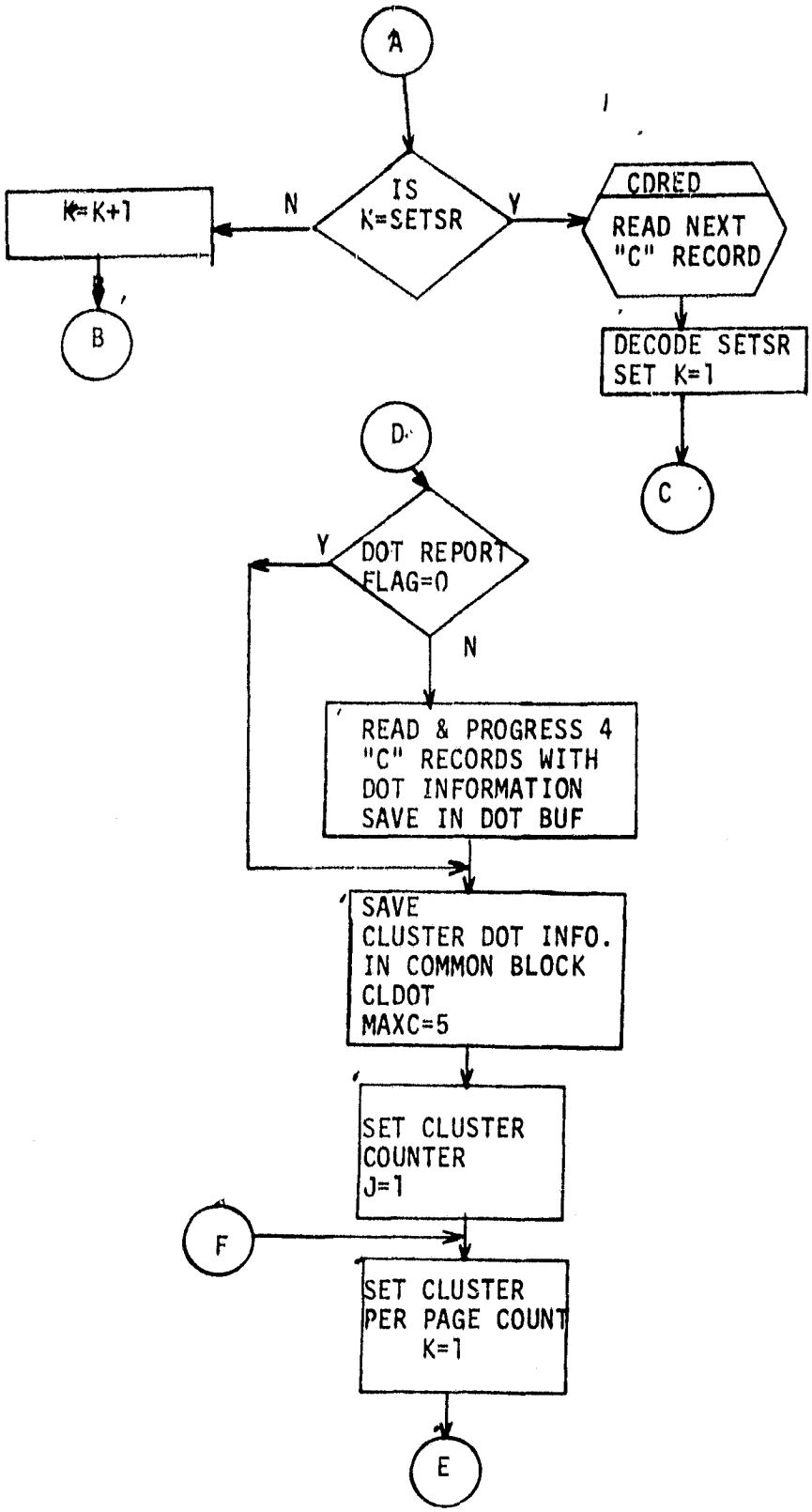
Flow diagram 1



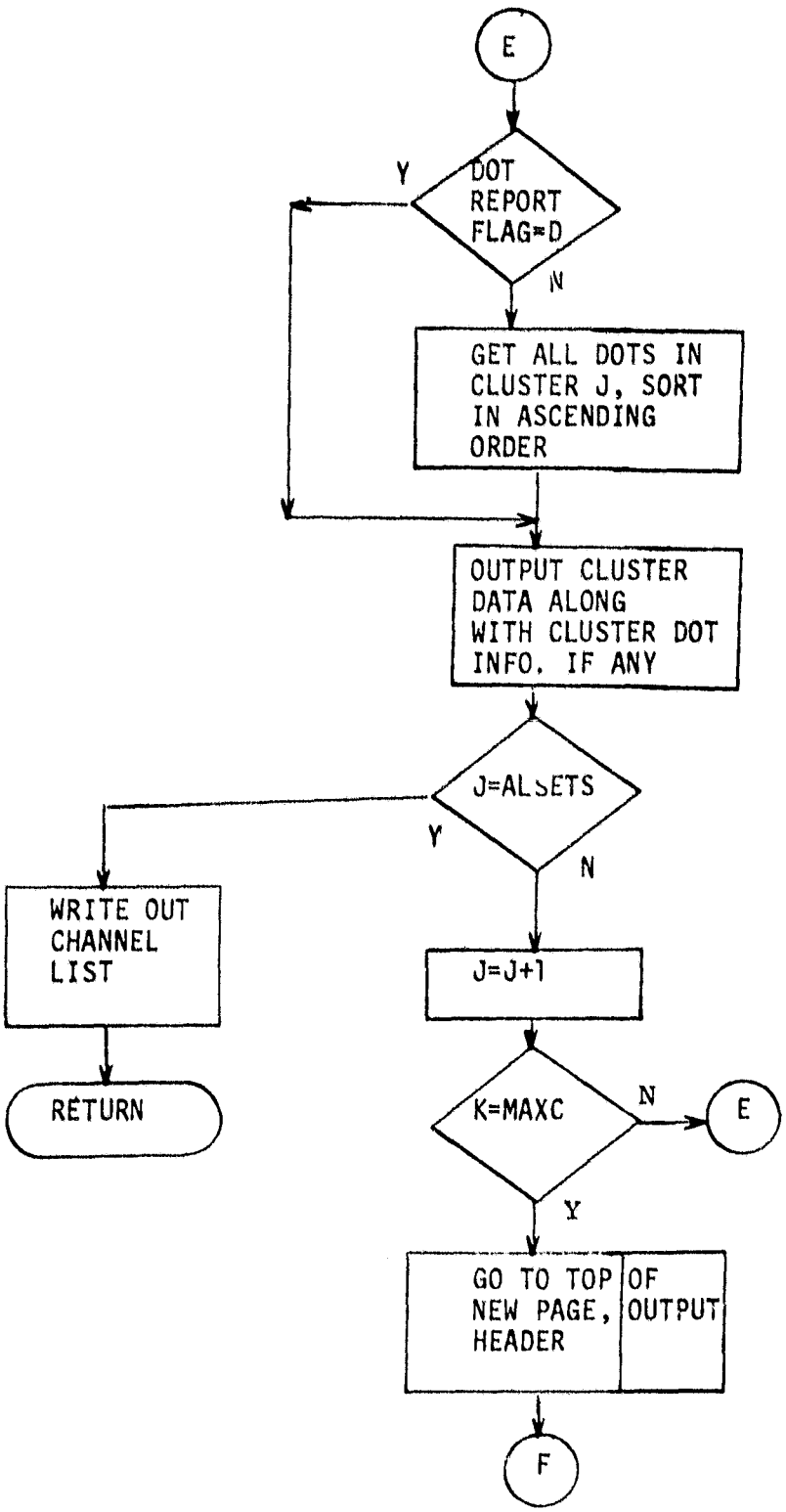
Flow diagram 1



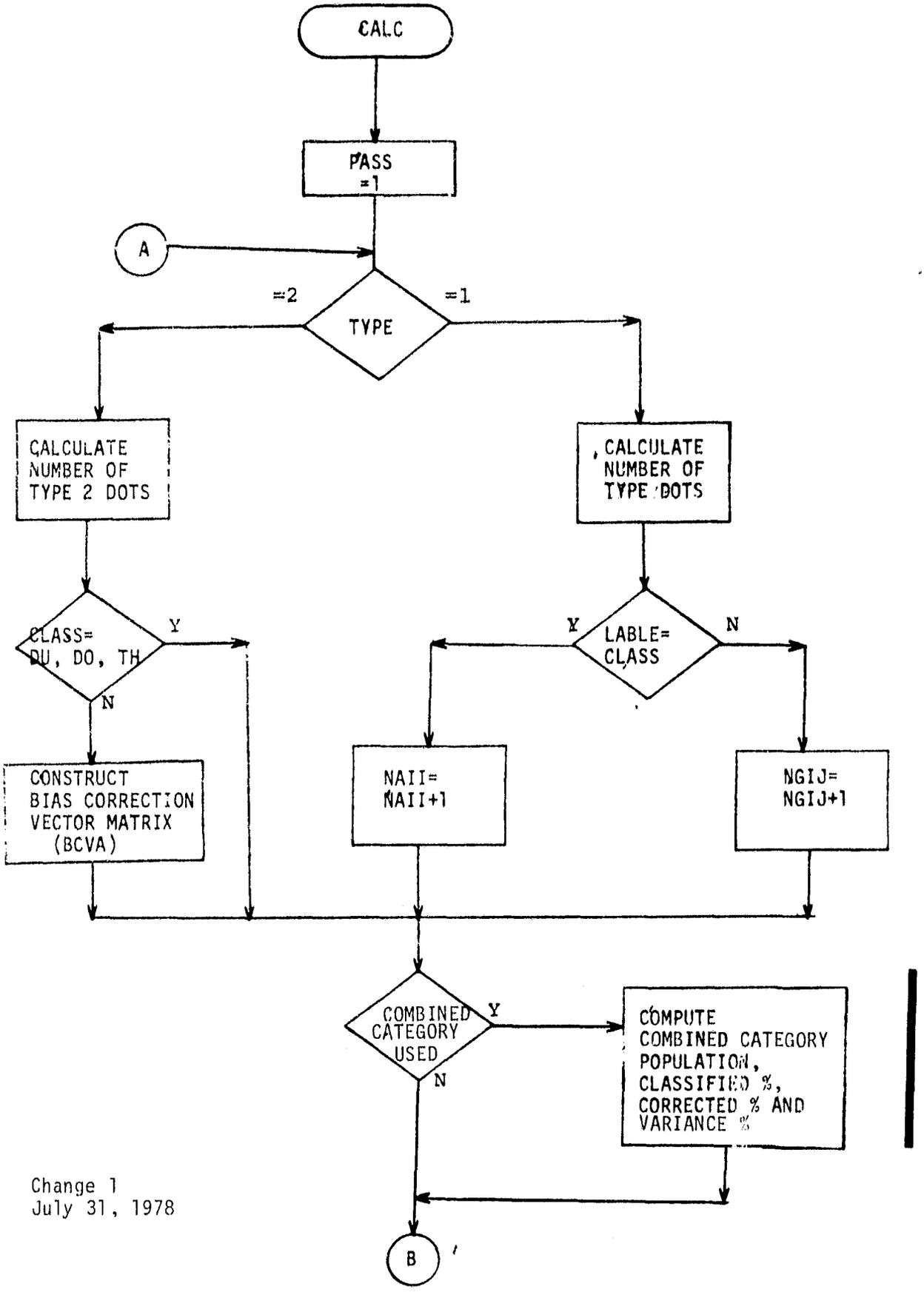
Flow diagram 2



Flow diagram 2

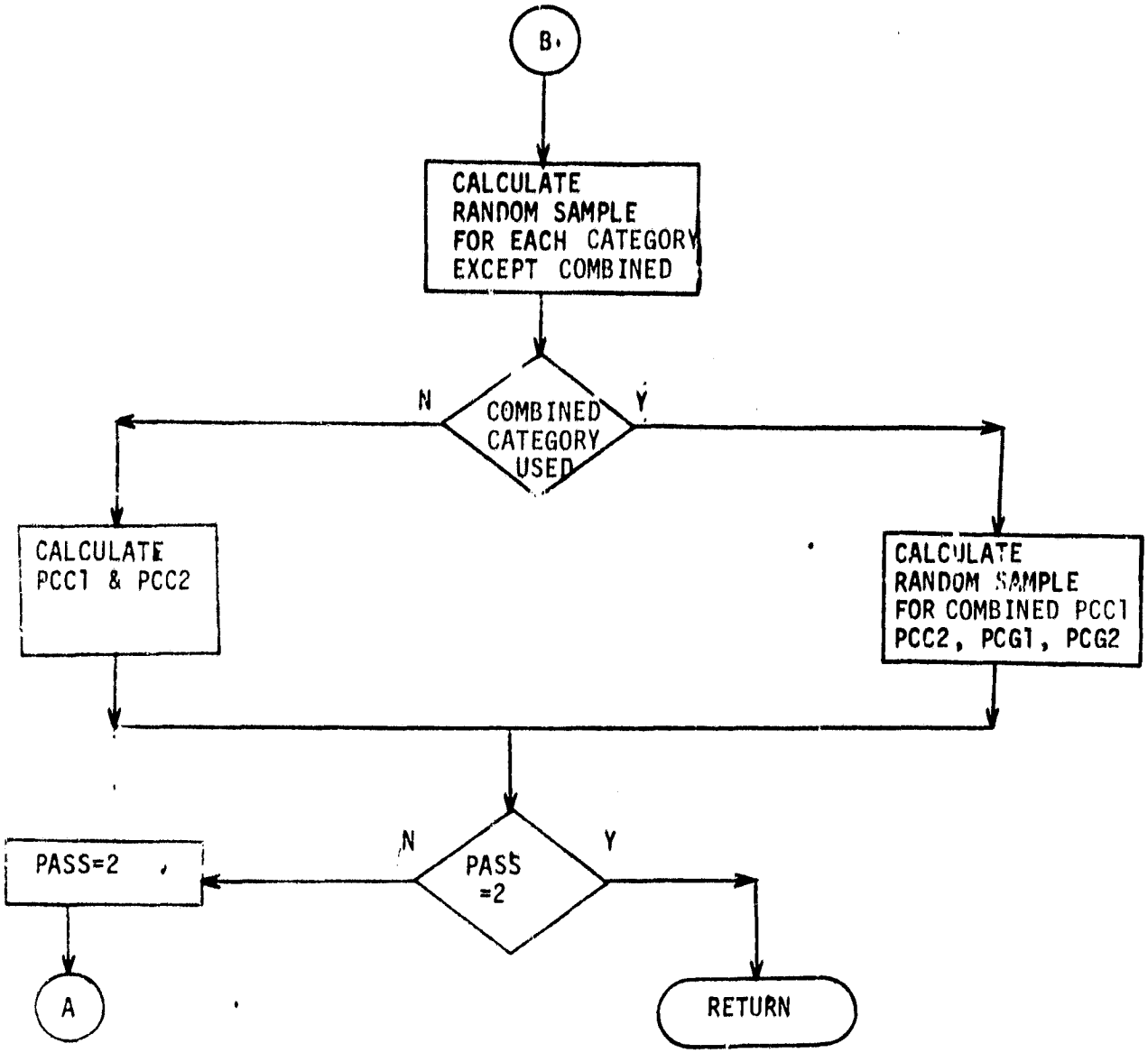


Flow diagram 2

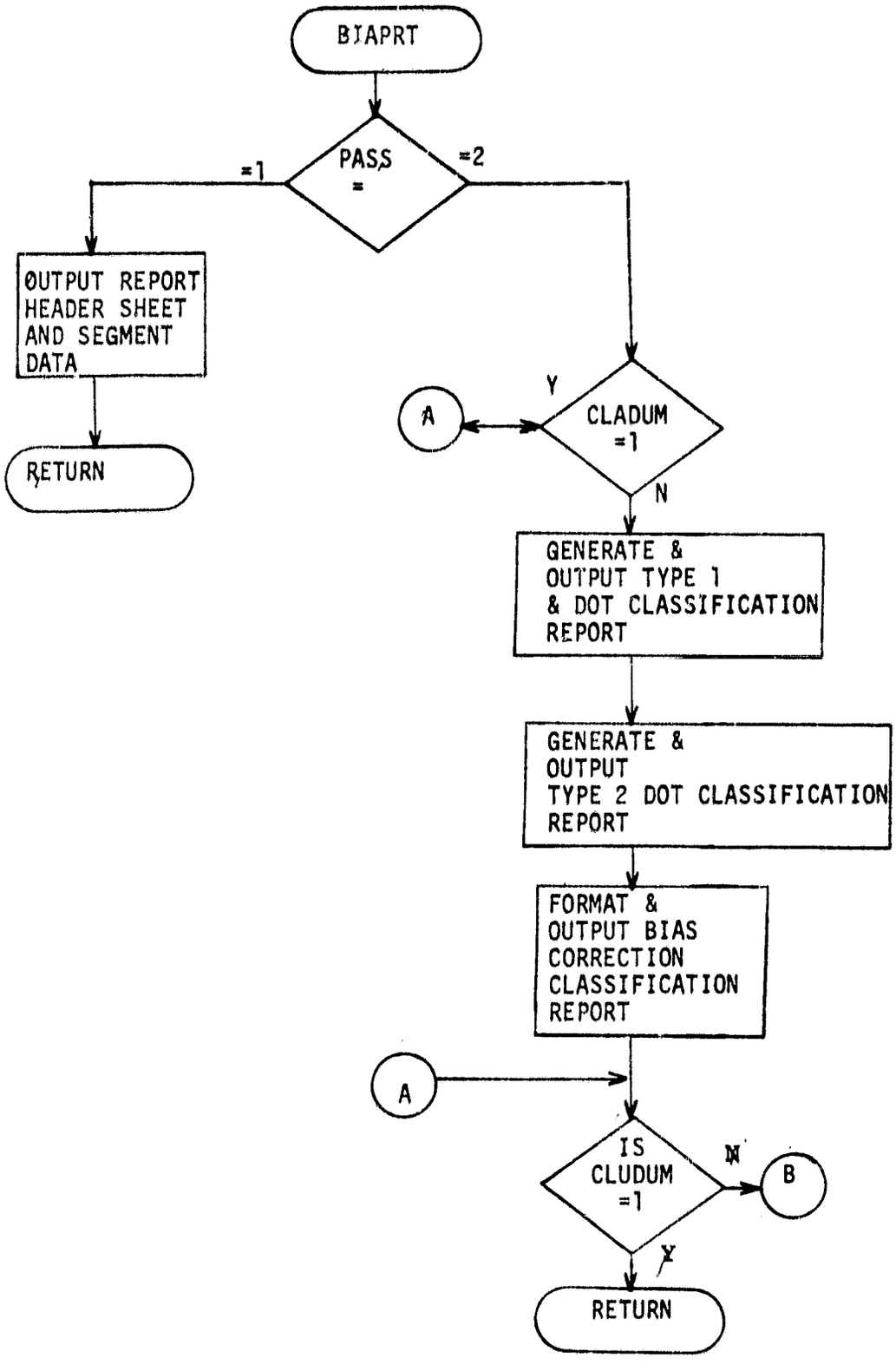


Change 1
July 31, 1978

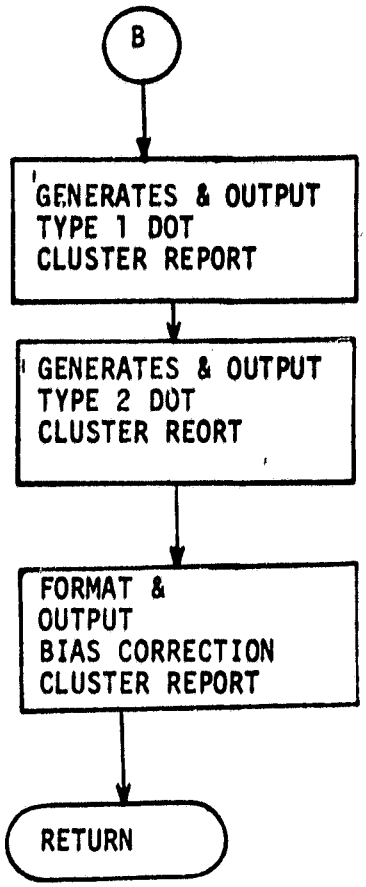
Flow diagram 3



THE LOGIC FOR SECOND PASS IS THE SAME AS THE FLOW CHART ONLY THE VARIABLES ARE CHANGED.



Flow diagram 4



Flow diagram 4

4. OPERATING PROCEDURE

4.1 GENERAL

CAMRPT is an RSX-11D Batch program which reads a CAMS/CAS Interface tape and generates a series of reports which are output on the line printer. It requires data card inputs.

4.2 DECK SET UP

The first 22 data cards define the input tape drive, input tape unit and curve fitting constants for DW & DS calculations. The DW & DS Formulae are shown in Appendix B. The card formats are:

M or X (tape drive)
0 or 1 (tape unit)
10 DW curve fitting constant cards as described in Appendix C
10 DS curve fitting constant cards as described in Appendix C
1 Combined category card as described in Appendix D.

Entries always start in column 1. To execute the CAMRPT default option for a limited printout of reports, an END card must follow the data cards above. If the option for a full output of all reports is desired, the control card sequence is:

```
A      (for all reports)
END
```

If the user desires to obtain the output for only a single segment on the input tape, segment 9681 for example, the control card sequence is:

```
S      9681
END
```

To obtain the output for segment 9681, and all segments following 9681, the control card sequence is:

```
S      9681
A
END
```

APPENDIX A

The above option is used when there is a bad segment on the input tape, to obtain the output for setments follcwing the bad seqment.

The Batch deck set up for the CAMRPT default option using input tape unit MTO is as follows:

```
$JOB/NAME=CAMRPT/MCR/LIMIT=99/ACCOUNT=50 50
$DATA
M
0
10 DW constant cards
10 DS constant cards
 1 combined category card
END
$EOD
$MCR REM RSXBAT
$RUN CAMRPT
$FOJ
```

To run the program, mount the CAMS/CAS Interface tape and enter a mount message.

For MTO the message would be:

```
MCR > MOU MTO:/CHA=[FOR]      (CR)
```

Then load the card reader with the CAMRPT Batch deck and enter BAT CR:, to read in the deck.

CALC.FTN /TR:1:1:CKS/WR

```
0001                SUBROUTINE CALC
0002                IMPLICIT INTEGER(S-Z)
0003                INCLUDE 'COMMON1.FIT'
0004 *              BYTE VARSEV(4*8),CLIENT(10)
0005 *              LOGICAL* TYPE(20),LABEL(200)
0006 *              LOGICAL* FACLA(120),SPCLA(120),RUCLAB(26),BULABL(26),GRAINS(26)
0007 *              REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
0008 *              REAL FACA,CLUC,TCCS,RCCS,CC7E
0009 *              REAL RATC2,OUTC2,BARC2 ,PURC2A
0010 *              DIMENSION CLAA(2),CLUC(2),TCCNS(2,8),CCCNS(2),RCCNS(2,8,8)
0011 *              DIMENSION RATC2(2,8),OUTC2(2,8),BARC2(2,8,8),PURC2A(2,8,8)
0012 *              DIMENSION CLAA(2,26),CLUC(2,26)
0013 *              DIMENSION BARP2(30),SUP P(20)
0014 *              COMMON/CRIAS/BARPP,SUPP,BALTC2,RUCKC2,RCCS,BUCAR,BACATN,BUCATN,
     *                  BARA2,BURA2,FARANS,BURANS,A,PC,PN,TC,BUCATS,BUCLAR,
     *                  OUTAPC,CLAAA,CLASSY,TYPE,CLIENT,FACATS,BACLAR,AUMC2,
     *                  PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,APRP,
     *                  CCNS,CCCNS,CLUC,RCCNS,CLAA,CLUC,TCCNS,RCCNS,CCCNS,
     *                  RATC2,OUTC2,BARC2,PURC2A,LABC,GRAINS
0015 *              COMMON/VARRY/CLAA,CLUC,TCCNS
0016 *              COMMON/CLCOM/CLPLT
0017 *              REAL BA(3,2)
0018 *              DIMENSION BALPC(30),BUPC(30),FACBR(30),BUCBR(30)
0019 *              REAL BACAT(20,2),BUCAT(20,26),BAVAR(30),BUVAR(30)
0020 *              REAL BARANS(30),BURANS(30)
0021                DIMENSION CNT(10),CNT(26)
0022                REAL PSLP,VAR,PI ,ALP,P/SE1,PASLP,A3F,BASEF
0023                REAL BASTE,ALPHF,PLSTE,PICNF,BAFF,BASF,ZZ
0024                REAL PCLP,LOATP(30),SLP,CLP,ARV(8),BARPPF(30)
0025                BYTE TY,BL,CT(2),CL2,CNT(26)
0026                LTYP2=0
0027                NTYP1=0
0028                WAIT=0
0029                VERY=0
0030                HIGH=0
0031                BASE=22942
0032                I=0
0033                BARPF(2H)=BPP
0034                BA(1,2)=BA(2,2)
0035                BYPE=0
0036                CHEK=0
0037                BASE=0
0038                BARP2(NA)=0
0039                BUPP(I)=0
0040                BARP2(NA)=0
0041                BALPC(NA)=0
0042                      BARANS(NA)=0
0043                NBL=0
0044                DO 5 I=1,2
0045            5   BA(I,I)=0
0046                N=0
0047                M=0
0048                S=0
0049                PCC1=0
0050                PCC2=0
0051                PCL1=0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
                        PRINT 1
                        PRINT 2
0053                    IF(CLADDR.EQ.1) GO TO 470
0054                    DO 15 I=1,MAXATS
0055                    IF(ATCLAR(I).EQ.'T') T=1
0056                    IF(ATCLAR(I).EQ.'S') S=1
0057                    IF(ATCLAR(I).NE.CLADDR) GO TO 15
0058                    IF(ATCLAR(I).GE.GRATES(J)) GO TO 10
0059                    GRATES(J)=ATCLAR(I)
0060                    SWITCH(NCLM)=I
0061                    GO TO 15
0062    10              CONTINUE
0063    12              IF(ATCLAR(I).NE.' ') GO TO 15
0064                    I=I
0065    15              CONTINUE
0066                    DO 4 I=1,11
0067                    LCATS(I)=0
0068    40              CONTINUE
0069                    DO 4 J=1,2
0070                    DO 4 I=1,2
0071                    CONV(J,I)=0
0072    401             CONTINUE
0073                    DO 4 I=1,31
0074                    DATA X(I)=0
0075    4               CONTINUE
0076                    DO 45 I=1,200
0077                    L=TYPE(I)
0078                    D=CLASSIFY(2*I-1)
0079                    C=CLASSIFY(2*I)
0080                    IF(L.EQ.' ') GO TO 3
0081                    IF(L.EQ.'1'.OR.L.EQ.'2'.OR.L.EQ.'3') GO TO 49
0082                    IF (TYPE(I).EQ.'2'.OR.TYPE(I).EQ.'0') GO TO 35
0083                    IF (C.EQ.' ') GO TO 45
0084
                C
                C COMPUTE TYPE 1 DOT DATA
                C
0085                    TYPE1=TYPE1+1
0086                    IF (C.EQ.'A'.OR.C.EQ.'B') GO TO 32
0087                    DATA=DATA+1
0088                    GO TO 35
0089    32              IF(C.LT.2)GO TO 35
0090                    DO 33 II=1,MAXAT
0091                    IF(C.NE.ATCLAR(GRATES(II))) GOTO 34
0092                    DO 33 JJ=1,2
0093                    IF(JJ.EQ.II) GOTO 33
0094                    IF(C.NE.ATCLAR(GRATES(JJ))) GOTO 33
0095                    DATA=DATA+1
0096    33              CONTINUE
0097    34              CONTINUE
                C
                C COMPUTE TYPE 2 DOT
                C
0098    35              IF (TYPE(I).EQ.'2'.OR.TYPE(I).EQ.'0') GO TO 45
0099                    IF(C.EQ.' ') GO TO 45
0100                    TYPE=TYPE+1
0101                    IF(C.EQ.'A'.OR.C.EQ.'B'.OR.C.EQ.'C') NCLC=NCLC+1
```

```
                C
                C CONSTRUCT  WAS CORRECTLY VERIFIED(LC)
                C
     0102               Z1=0
     0103               Z2=
     0104               D  I=1,NCATS
     0105               IF(C  .EQ.FALLA(I))I=
     0106               IF(C  .EQ.BACLAD(I))Z2=
     0107     43        C  I=1  E
     0108               IF(Z1.EQ.0..OR.Z2.EQ.0.)GO TO 39
     0109               CORA(I,Z2)= CVA(Z1,Z2)+1
     0110     39        C  I=I  IF
                C
                C COMPUTE CROSS POPULATION, CLASSIFIED %, CORRECTED %
                C
     0111               IF(NCLUM.LT.2)GO TO
                C
                C COMPUTE  ERROR CORRECTED % AND VARIANCE %
                C
     0112               ALCT=
     0113               DO  II=1,NCLU
     0114               PA  (A)=CAP.P(A)+ AP.P(SHALPL(II))
     0115               CAT  (A)=C  C(A)+CATACT(CRNTPL(II))
     0116               DO  JJ=1,NCLU
     0117     55        AL  =ALGT+CVA(CRNTPL(II),CRNTPL(JJ))
     0118     60        CONTI UE
     0119               ALCT=
     0120               DO  I=1,NCATS
     0121               DO  II=1,NCLU
     0122     95        ALCT=ALGT+CVA(I,CRNTPL(II))
     0123     98        CONTI UE
     0124               ALCT=
     0125               ALCTF=FLOAT(ALGT)
     0126               ALCF=FLOAT(ALCF)
     0127               IF(ALCF.GT.0) ALCT=ALCTF/ALCF
     0128               ALVA=
     0129               ALCT=
     0130               DO 100 I=1,NCATS
     0131               DO 110 II=1,NCLU
     0132               IF(I.EQ.SHALPL(II)) GO TO 125
     0133    110        CONTI UE
     0134               DO 120 J=1,NCATS
     0135               DO 120 II=1,NCLU
     0136               IF(J.EQ.SHALPL(II)) GO TO 125
     0137    120        CONTI UE
     0138               ALCT=ALCT+CVA(I,J)
     0139    125        CONTI UE
     0140               DO 140 I=1,NCATS
     0141               IF(CATACT(I).EQ.0) GO TO 140
     0142               DO 130 II=1,NCLU
     0143               IF(I.EQ.CRNTPL(II)) GO TO 140
     0144    130        CONTI UE
     0145               ALVA=ALVA+CVA(I,J)
     0146    140        CONTI UE
     0147               CATACT=FLOAT(NSET-NCLU-Z1-Z2)
     0148               CATACT=0.
```

```
0149                  RAPF=ALOG10(RAPF+1.E-5)
0150                  IF(BASEF.GT.0.) BASE1=BAPF/BASEF
0151                  ALNTF=FLOAT(ALNT)
0152                  ALNBF=FLOAT(ALNB)
0153                  N3F=0
0154                  IF(ALNBF.GT.0) AL=ALNTF/ALNBF
0155                  N3F=0.
0156                  DO 155 I=1,BACATS
0157                  DO 155 II=1,NGLM
0158                  IF(I.EQ.GRNLBL(II)) GOTO 159
0159     155          CONTINUE
0160                  N3F=N3F+FLOAT(RAP2P(I))
0161     159          CONTINUE
0162                  BACKP(MA)=0.
0163                  BASE2=0.
0164                  IF(BASEF.LE.0) GO TO 161
0165                  RAPCNT(MA)=(ALOG(BASE1+1.)-ALN)*N3F/BASEF)*100.
0166                  BASE2=(N3F/BASEF*100.)**2
0167     161          Z4=0
0168                  DO 21 J=1,RACATS
0169                  DO 21 II=1,NGLM
0170     20           Z4=Z4+RCVA(J,GRNLBL(II))
0171     21           CONTINUE
0172                  VAR=0.
0173                  DO 22 I=1,NTAT(Z4=1)
0174                  IF (Z4.LE.1) GO TO 24
0175                  VAR=((BASE1+1.)**2)*(ALOG(1+(I-ALG)))/Z4F
0176     24           IF(ALNBF .GT. 1.) GO TO 22
0177                  RAVAR(MA)=VAR
0178                  GO TO 25
0179     22           AVA (MA)=VA+VAR*Z4*(I-ALN)/(ALNBF-1.)
                C
                C COMPUTE THE RANDOM SAMPLE FOR EACH CATEGORY EXCEPT FOR GRAIN
                C
0180     25                 RATOT=0
0181                  PSUM=0.
0182                  DO 167 I=1,BACATS
0183                  PSUM=PSUM+FLOAT(RAP2P(I))
0184     167          CONTINUE
0185                  PSUM=PSUM/(2.5*2.,-RAP2P(NAI))
0186                  ANTF=FLOAT(NTOT)
0187                  DO 165 I=1,BACATS
0188                  RCVT(I)=0
0189                  DO 165 J=1,BACATS
0190                  RCVT(I)=RCVT(I)+RCVA(I,J)
0191     165          CONTINUE
0192                  RARAS(I)=0.
0193                  IF(NTOT.GT.1) RARAS(I)=FLOAT(RCVT(I))/ANTOT*PSUM*100.
0194                  RAT=RAT+RCVT(I)
0195                  DO 163 II=1,NGLM
0196     163          IF(I.EQ.GRNLBL(II)) GNR=GNR+RCVT(I)
0197     165          CONTINUE
                C
                C COMPUTE THE RANDOM SAMPLE FOR GRAIN
                C
0198                  IF(NGLM.LT.2)GOTO 130
```

```
0199                GWSF=FLOAT(GWS)
0200                IF(NCLF.GT.0) GARANS(KA)=GWSF/ACLF*PSIM*100.
                C
                C COMPUTE PCC1, PCC2, PCCG1, PCCG2
0201          100   DATAT=0
0202                X=0
0203                IF (NTYP1.EQ.0) GO TO 45
0204                PCC1=FLOAT(NAII)*100./FLOAT(NTYP1)
0205          45    IF(NTYP2.EQ.0) GO TO 465
0206                DO 50 I=1,6
0207          50    X=X+PCVA(I,I)
0208                PCC2=FLOAT(X)*100./FLOAT(NTYP2)
0209          465   IF(NUM.LT.2)GO TO 470
0210                DO 174 II=1,NGLB
0211                DO 173 JJ=1,NGLB
0212                IF(II.EQ.JJ) GO TO 173
0213                DATAT=DATAT+PCVA(GRGLBL(II),GRGLBL(JJ))
0214          173   CONTINUE
0215          174   CONTINUE
0216                DO 175 I=1,NACATS
0217                DATAT=DATAT+PCVA(I,I)
0218          175   CONTINUE
0219                IF (NTYP1.EQ.0)GO TO 468
0220                PCCG1=(FLOAT(NAII)+FLOAT(NCIJ))*100./FLOAT(NTYP1)
0221          468   IF(NTYP2.EQ.0)GO TO 47
0222                PCCG2=FLOAT(NATAT)*100./FLOAT(NTYP2)
0223          47    CONTINUE
                C CLEAR CALCULATIONS FIRST
0224                DO 230 MM=1,2
0225                IF(MM.EQ.1) KK=
0226                IF(MM.EQ.2) KK=5
0227                SL=0.
0228                CL=0.
0229                DO 240 LL=1,2
0230                DO 230 II=1,6
0231                DATC(II,LL)=0.
0232                DATP(II,LL)=0.
0233                DO 230 LI=1,6
0234                DATP(II,LL,LI)=0.
0235                DUPC(II,LL,LI)=0.
0236          230   CONTINUE
0237                IF(MM.EQ.0) GO TO 270
0238                NARY(7)=0.
0239                NARY(1)=ARCC(KK)
0240                NARY(2)=NUMCC(KK)
0241                NARY(3)=DATA1S(KK)
0242                NARY(4)=NATAT(KK)
0243                NARY(5)=NCAIN(KK,KK)
0244                NARY(6)=0.1
0245                IF(NCAIN.GT.1) NARY(7)=NCAIN(K,K)
0246                NARY(8)=PCC2
0247                DO 250 I=1,8
0248                DATC(MM,I)=TRANS(MM,I)*NARY(I)
0249                SL=SL+DATC(MM,I)
0250                DO 250 J=1,6
0251                DARC(MM,I,J)=TRANS(MM,I,J)*NARY(I)*NARY(J)
```

```
0252                    CL =CL+ARC*COS(C**2*J)
0253        240         CONTINUE
0254        250         CONTINUE
0255                    SL =SL+CORKS(MA)
0256        270         CLAR(MA)=CL*SIN
0257        290         CONTINUE
0258                    ITYPE=0
0259                    NA11=0
0260                    NP11=0
0261                    NG1J=0
0262                    BASE=02932
0263                    GAS=0
0264                    DOF P(KU)=0
0265                  BUVAR(KU)=0
0266                    BUC*R(WU)=0
0267                    BUONC*(WU)=0
0268                          BURA'S(WU)=0
0269                    NGLM=0
0270                    DO 9 0 I=1,25
0271        900         GRN'N(I)=0
0272                    X=0
0273                    S=0
0274                    PCL1=0
0275                    NPCL=0
0276                    ITYPI=0
0277                    PCL2=0
0278                    PRLC1=0.
0279                    PCL02=0.
0280                    IF(CL.EQ.0.0) RETURN
0281                    DO 915 I=1,LOCATS
0282                    IF(ALA(I).EQ.'*') S=1
0283                    IF(ALA(I).EQ.'S') S=1
0284                    DO 91; J=1,NLACO
0285                    IF(ALA(I).NE.BEA' S(J)) GO TO 910
0286                    NGLM=NGLM+1
0287                    GRN'N(NGLM)=1
0288                    GTT' 915
0289        910         CONTINUE
0290        915         CONTINUE
0291                    DO 94; I=1,11
0292                    LOCAT(I)=0
0293        94A         CONTINUE
0294                    DO 940 J=1,26
0295                    DO 941 I=1,26
0296                    RCV(J,I)=0
0297        9401          CONTINUE
0298                    DO 941 I=1,3?
0299                    CUR'A S(I)=0
0300        941         CONTINUE
0301                    DO 97A I=1,2?6
0302                    LA=LOCAT(I)
0303                    CL=CL'AT(2*I-1)
0304                    CLC=CL'AT(2*I)
0305                    IF(I .EQ. 1 ) GO TO 929
0306                    IF (CL .EQ. TYPE .AND. CLC.EQ.TYPE) GO TO 939
0307                    IF (TYPE(I).NE.'2'. R.TYPE(I).EQ.'0') GO TO 935
```

```
0308                  IF (CL2.EQ.' ') GO TO 939
                C
                C COMPUTE TYPE 1 DAT DATA
                C
0309                  NTYP1=NTYP1+1
0310                  IF (CL.NE.CL1.AND.CL.NE.' ') GO TO 932
0311                  VALUE(II)=1
0312                  GO TO 935
0313            934   IF(CL1.LT.2)GO TO 935
0314                  DO 936 II=1,NCL
0315                  IF(CL.NE.BYCLASS(TRNCL(II))) GOTO 936
0316                  DO 937 JJ=1,NGRN
0317                  IF(JJ.EQ.II) GO TO 938
0318                  IF(CL.NE.BYCLASS(GRNCL(JJ))) GOTO 937
0319                  VALUE(II)=1
0320            937   CONTINUE
0321            936   CONTINUE
                C
                C COMPUTE TYPE 2 DATA
                C
0322            935   IF (TYPE(1).EQ.'(') GO TO 939
0323                  IF(CL2.NE.' ') GO TO 939
0324                  NTYP2=NTYP2+1
0325                  IF(CL2.NE.' '.AND.CL.EQ.' ') NCL=NCL+1
                C
                C CONSTRUCT BIAS CORRECTION VECTORS(LB)
                C
0326                  Z1=0
0327                  Z2=0
0328                  DO 944 K=1,NCATS
0329                  IF(CL.EQ.BYCLASS(K))Z1=1
0330                  IF(CL2.EQ.BYCLASS(K))Z2=1
0331            943   CONTINUE
0332                  IF(Z1.EQ.0.OR.Z2.EQ.0)GO TO 939
0333                  CVL(Z1,Z2)=CVL(Z1,Z2)+1
0334            939   CONTINUE
                C
                C COMPUTE GRAIN POPULATION CLASSIFIED %, CORRECTED %
                C
0335                  IF(NGRN.LT.2)GO TO 925
                C
                C COMPUTE GRAIN CORRECTED % AND VARIANCE %
                C
0336                  ALGT=0
0337                  DO 941 II=1,NGRN
0338                  BUP=BUP+VALUE(II)*BYCLASS(GRNCLBL(II))
0339                  BJUP=BJUP+VALUE(II)*BYCLASS(GRNCLBL(II))
0340                  DO 945 JJ=1,NCL
0341            945   ALGT=ALGT+CVL(GRNCLBL(II),TRNCLBL(JJ))
0342            941   CONTINUE
0343                  ALGT=0
0344                  DO 942 I=1,NCATS
0345                  DO 943 II=1,NCL
0346            943   ALGT=ALGT+CVL(I,BYCLASS(II))
0347            942   CONTINUE
0348                  ALGT=FLOAT(ALGT)
```

```
CALC.FTN          /FREQ:C:S/I:S

0349                AL=FLOAT(ALPC)
0350                ALG=0.
0351                IF(ALGBF.GT.0) ALG=ALGTF/ALGBF
0352                ALPC=0
0353                ALT=0.
0354                DO 9125 I=1,NUCATS
0355                DO 9120 II=1,NVL
0356                IF(J.EQ.GRPL(II)) GOTO 9125
0357      9120      CONTINUE
0358                DO 9125 J=1,NUCATS
0359                DO 9190 II=1,NVL
0360                IF(J.EQ.GRPL(II)) GOTO 9125
0361      9190      CONTINUE
0362                ALT=ALT+PCVU(I,J)
0363      9125      CONTINUE
0364                DO 9150 I=1,NUCATS
0365                DO 9150 J=1,NUCATS
0366                DO 9145 II=1,NVLM
0367                IF(J.EQ.GRPL(II)) GOTO 9150
0368      9145      CONTINUE
0369                ALPI=ALPI+PCVU(I,J)
0370      9150      CONTINUE
0371                BASEF=FLOAT(BASE-NUMPP(NUM))
0372                BASE1=0.
0373                BAPF=FLOAT(NUMPP(NUM))
0374                IF(BASEF.GT.0) BASE1=BAPF/BASEF
0375                ALTTF=FLOAT(ALPT)
0376                ALPIF=FLOAT(ALPI)
0377                ALG=0.
0378                IF(ALGBF.GT.0) ALG=ALGTF/ALGBF
0379                SP=0.
0380                DO 9159 I=1,NUCATS
0381                DO 9155 II=1,NVLM
0382                IF(I.EQ.GRPL(II)) GOTO 9159
0383      9155      CONTINUE
0384                SP=SP+FLOAT(NUMPP(I))
0385      9159      CONTINUE
0386                DO 9161 I=1,NUCATS
0387                BASE2=0.
0388                IF(BASEF.LE.0) GO TO 9161
0389                PCP(I)=(ALOG(BASE1+(1.-ALN)*NBF/BASEF)*100.
0390                BASE2=(NBF/BASEF*100.)**2
0391      9161      END
0392                DO 924 J=1,NUCATS
0393                DO 923 II=1,NVLM
0394      923       SA=SA+PCVU(J,GRPL(II))
0395      924       CONTINUE
0396                VAR=0.
0397                ANT=FLOAT(SA-1)
0398                IF (SA.LE.1) GO TO 926
0399                VAR=((BASE1*100.)**2*(ALOG(1.-ALG)))/SAF
0400      926       IF(NBF .GT. 1.) GO TO 922
0401                PCVAR(I)=VAR
0402                GO TO 925
0403      922       DO 921 I=1,NUCATS PCVAR(I)=VAR*(1.-ALN)/(FLOAT(NBF-1.)
0404      925       C
```

```
CALC.FTN        /TYPE:TCMS/ST
C
C COMPUTE THE RANDOM SAMPLE FOR EACH CATEGORY EXCEPT FOR GRAIN
C
0404    925             BUTTYE=0
0405                    PSUM=0.
0406                    DO 9140 I=1,NUCATS
0407                    PSUM=PSUM+FLOAT(NCROP(I))
0408    9140            CONTINUE
0409                    PSUM=PSUM/(22272.*NUCROP(NAI))
0410                    NOBCLF=FLOAT(NOBCL)
0411                    DO 9145 I=1,NUCATS
0412                    ECVT(I)=0
0413                    DO 9145 J=1,NUCATS
0414                    ECVT(I)=ECVT(I)+CV(I,J)
0415    9145            CONTINUE
0416                    DO 9148 I=1,NOBCL
0417                    IF(NOBCLF.GT.0.) RUNRAN(I)=FLOAT(ECVT(I))/NOBCLF*PSUM*100.
0418                    BUTTYE=BUTTYE+ECVT(I)
0419                    DO 9143 I=1,NOBCL
0420    9143            IF(I.EQ.1.OR.RUNRAN(I).LT.RUNRAN(I-1)) RUNRAN(I)=RUNRAN(I)+ECVT(I)
0421    9148            CONTINUE
C
C COMPUTE THE RANDOM SAMPLE FOR GRAIN
C
0422                    IF(NOBCLM.LT.2)GOTO 9100
0423                    GRASF=FLOAT(9.5)
0424                    RUNRAN(NM)=0.
0425                    IF(NTOBCLF.GT.0.) RUNRAN(NM)=GRASF/NOBCLF*PSUM*100.
C
C COMPUTE PCT1, PCT2, PCT31, PCT32
0426    9100            PAT1TE=0
0427                    X=
0428                    IF ( NTYP1 .EQ. 0 ) GO TO 945
0429                    PCT1=FLOAT(NUIT)*100./FLOAT(NTYP1)
0430    945             IF( NTYP2 .EQ. 0 ) GO TO 9469
0431                    DO 9500 I=1,9
0432    9500            X=X+CVU(I,I)
0433                    PCT2=FLOAT(X)*100./FLOAT(NTYP2)
0434    9469             IF( NOBCLM.LT.2)GO TO 9470
0435                    DO 9173 II=1,NOBCLM
0436                    DO 9173 JJ=1,NOBCLM
0437                    IF(II.EQ.JJ)GOTO 9173
0438                    PATATE=PATATE+ECVU(GRANLBL(II),GRANLBL(JJ))
0439    9173            CONTINUE
0440    9174            CONTINUE
0441                    DO 9175 I=1,NUCATS
0442                    PATATE=PATATE+ECVU(I,I)
0443    9175            CONTINUE
0444                    IF (NTYP1.EQ.0)GO TO 9468
0445                    PCT31=FLOAT(NUIT+PATATE)*100./FLOAT(NTYP1)
0446    9468             IF( NTYP2.EQ.0)GO TO 9471
0447                    PCT32=FLOAT(PATATE)*100./FLOAT(NTYP2)
0448    9470            CONTINUE
C*****CALCULATE KAPPA STATISTIC
0449                    DO 9010 K=1,2
0450                    IF(K.EQ.1) GO
0451                    IF(K.EQ.2) GOES
```

```
0452                    SL=0.0
0453                    CL=0.0
0454                    IF(NPTS.EQ.0) GO TO 9270
0455                    WARY(I)=0.0
0456                    WARY(I)=SIN(XK)
0457                    WARY(I)=COS(XK)
0458                    WARY(I)=SIN(XK)
0459                    WARY(I)=COS(XK)
0460                    WARY(I)=SIN(XK)
0461                    WARY(I)=PCI2
0462                    IF(I.EQ.1) WARY(I)=SIN(XK)
0463                    WARY(N)=PCI2
0464                    DO 9250 I=1,N
0465                    PLTCOS(XK,I)=TCOS(XK,I)*WARY(I)
0466                    SL=SL+PLTCOS(XK,I)
0467                    DO 9250 J=1,N
0468                    PLTSIN(XK,I,J)=PSIN(XK,I,J)*WARY(I)*WARY(J)
0469                    CL=CL+PLTSIN(XK,I,J)
0470    9250            CONTINUE
0471    9260            CONTINUE
0472                    SL=SL+COS(XK)
0473    9270            CL(K)=CL+SL
0474    9280            CONTINUE
0475                    CONTINUE
0476                    RETURN
0477                    END
```

PRIMARY ENTRY POINTS

NUMBER	NAME	TYPE	ATTRIBUTES
1	[illegible]	[illegible] 2675	[illegible],LCL
2	[illegible]	[illegible] 2	[illegible],LCL
4	[illegible]	[illegible] 2432	[illegible],LCL
5	[illegible]	[illegible] 22	[illegible],LCL
6	CELSIUS	226743 4A72	[illegible],GRL
7	[illegible]	[illegible] 2	[illegible],GRL
8	CLOCK	[illegible] 206	[illegible],GRL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CALC		4=000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALG	[illegible]	[illegible]	ALGA	Ie2	[illegible]	ALGAF	Re4	[illegible]	ALGT	Ie2	[illegible]	ALGTF	Re4	[illegible]
AL	Ie1	4=000172	AL'	Ie2	4=001274	ALPF	Re4	4=000156	ALAT	Ie2	4=000776	ALATF	Re	=000152
AR	Ie1	[illegible]	A	Ie2	[illegible]	BACATS	Ie2	4=000232	BARE	Re4	4=000172	BASE	Ie2	4=000726
BASE	[illegible]	[illegible]	BASE4	Re4	4=000132	BASE2	Re4	4=000136	BATAT	Ie2	4=001004	BATCT	Ie2	6=023342
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	CELSIUS	Ie2	7=000042
CL	[illegible]	[illegible]	CL2	Le1	4=[illegible]	D	Ie2	6=014200	DU	Ie2	4=014252	GMS	Ie2	4=000732
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	J	Ie2	4=000754
JJ	Ie2	4=00[illegible]	K	Ie2	4=[illegible]	KK	Ie2	4=001012	LD	Le1	4=000642	LI	Ie2	4=001030
LL	Ie2	[illegible]	[illegible]	Ie2	[illegible]	[illegible]	Ie2	4=000744	MAIT	Ie2	4=000720	MDIT	Ie2	3=000722
MOIJ	Ie2	4=00[illegible]	MOL'	Ie2	4=[illegible]	MLARG	Ie2	4=026706	MMOD	Ie2	4=001042	MMOLF	Re4	4=000206
[illegible]	Ie2	[illegible]	[illegible]	Ie2	4=000714	[illegible]	Re4	4=001052	R	Ie2	4=001052	PMOA1	Re2	6=01A100
PMOA2	Re1	4=01[illegible]	PMO'	Re4	4=[illegible]	PMOD	Re4	6=16074	PMLOA	Re4	4=016120	PMLP2	Re4	6=01A124
PMLI	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	6=000112	S	Ie2	4=000750	SLW	Re4	4=000402
TC	Ie1	4=01[illegible]	UNI	Ie2	4=[illegible]	VAR	Re4	4=000116	W	Ie2	4=000746	WA	Ie2	4=000736
WI	Ie1	4=000[illegible]	X	Ie2	4=01[illegible]	Z1	Ie2	4=000752	Z2	Ie2	4=000744	Z4	Ie2	4=001000
Z45	Ie1	4=000[illegible]												

ARRAYS

NAME	TYPE	ADDRESS	TYPE	DIMENSIONS
A	[illegible]	[illegible]	[illegible]	(2,2)
[illegible]	[illegible]	[illegible]	[illegible]	(2,22)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	(22)
[illegible]	[illegible]	[illegible]	[illegible]	(22)
[illegible]	[illegible]	[illegible]	[illegible]	(2)
[illegible]	[illegible]	[illegible]	[illegible]	(2)
[illegible]	[illegible]	[illegible]	[illegible]	(30)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	(2,2)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
5	**	17	**	19	**	25	1=000804	28	**
21	**	22	1=003722	24	1=003722	25	1=003454	32	1=001104
33	1=001240	34	1=001260	35	1=001316	39	1=001414	40	**
41	**	44	**	45	1=004174	50	**	55	**
60	**	85	**	95	**	100	1=004124	115	**
120	**	125	1=002260	130	**	140	1=002454	155	**
159	1=002564	160	**	171	1=003414	173	**	185	**
188	**	175	1=001	174	**	175	**	239	**
240	**	270	**	272	1=003634	280	**	401	**
408	1=004300	409	1=004710	470	1=004444	900	**	910	1=000674
940	1=001210	921	**	921	**	922	1=001104	924	1=001140
925	1=001170	942	1=001	943	1=000754	944	1=000774	945	1=000720
946	1=000	947	**	961	**	983	**	965	1=001112
9	**	965	**	960	**	995	**	994	**
9101	1=001	9115	**	9120	**	9130	1=000000	9145	**
9150	1=000171	9155	**	9160	1=001	9165	**	9161	1=000634
9167	**	9165	**	9160	**	9170	1=002150	9174	**
9175	**	9176	**	9150	**	9201	1=001	9200	**
9201	**	9202	1=002	9200	1=001	9207	1=002		

TOTAL SPACE ALLOCATED = 041100 9248

.LPI=CALC

```
FORTRAN IV-PLUS V02-01          11:51:33    26-JUL-78          PAGE 1
BIAPRT,FTN       /FR:1:FORS/WT
                                                                                1
                                                                                1
0001            SU[illegible] TITLE BIAPRT(INPUT,TAPE8,PASS)
0002            IMPLICIT INTEGER (A-Z)
0003            LOGICAL*1 IB(1)
0004           1,OPEN 'PCL',1,'AT'
0005 *          BYTE A ASSY(4,26),'LABEL'(418)
0006 *          LOGICAL*1 TYPE(26),LBLED(209)
0007 *          DIMENSION PCL(2,2),RACOR(26),RUCLAB(26),BULABL(26),GRAINS(26)
0008 *          REAL*8 PCL1,PCL2,PCLG1,PCLG2
0009 *          REAL CLAD,CLUS,TOTS,RCOS,CCOS
0010 *          REAL ATOT(), TOTS(),RCOS,RUROS
0011 *          DIMENSION CLAD(2),CLUS(2),TOTS(2,8),CCOS(2),RCOS(2,8,8)
0012 *          DIMENSION PARCO(2,8), TOTS(2,8),PARCOS(2,8,8),RUROS(2,8,8)
0013 *          DIMENSION SCOR(2,2,8),PCOL(26,26)
0014 *          DIMENSION PARCO(30), UPCR(30)
0015 *          COMMON /CRIAS/ PCT,RTP,BALCOR,RUUNCR,BACOR,BUCOR,BACAIN,RUCAIN,
     *                  NAME,  AR,TARANS,BGRANS,A,DA,DU,TC,BUCATS,RUCLAB,
     *                  BULABL,LABEL,CLASFY,TYPE,LBLED,RACATS,BACLAB,AUNCR,
     *                 PCT1,PCOR,PCLG1,PCLCR,PCL1,PCL2,PCLG1,PCLG2,APTR,
     *                  TOTA,BCOR,RATE,PCTCT,CLAD,CLUS,TOTS,RCOS,CCOS,
     *                  ATOT,TOTS,PARCOR,RUROS,LABG,GRAINS
0016 *          COMMON /PRAV/CLAD,CLUS
0017 *         COMMON /PCLCOM/PCL
0018 *          REAL A(3,2)
0019 *          DIMENSION PARCOR(30),BUCOR(30),RACOR(30),RUCOR(30)
0020 *          REAL*8 PCOL(26,26),BAVAR(30),BUVAR(30)
0021 *          REAL PARAVR(30), UPAR(30)
0022            COMMON /SPEC/SPEC(2
0023            LOGICAL*1 SPEC(1),P(120),BLANK,PMATR(120)
0024           DIMENSION PCL(10)
0025            DATA PCL/10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     *                 150,160,170,180,190/
0026            DATA BLANK/' '/,A/'1'/,SLASH/'/'/,TWO/'2'/
0027       100  FORMAT(1H1,44X,' IRIS CORRECTION CLASSIFICATION REPORT')
0028       101  FORMAT(1HO,2X,1(5X,'CAT'))
0029       102  FORMAT(1H ,12X,1(I2,3X))
0030       103  FORMAT(1HO,5X,'# PLOTTED      ',10(3X,I5))
0031       104  FORMAT(1H ,5X,'% OF FIELD     ',10(2X,I6))
0032       105  FORMAT(1H ,5X,'% OF TOT       ',10(2X,I6))
0033       106  FORMAT(1H ,5X,'% OF CF        ',10(2X,F6.2))
0034       107  FORMAT(1H ,5X,'% OF SAMPLE    ',10(2X,F6.2))
0035       108  FORMAT(1H1,51X,'CLASS VALUE MATRIX')
0036       109  FORMAT(1H ,14X,(I3)' CAT',I3))
0037       110  FORMAT(1H ,1X,'CAT',  F I3,7(13(6X,1A1,1X))
0038       111  FORMAT(1H ,1X,'CAT',  F I3,')
0039       112  FORMAT(1H ,11X,1-13(,13(4X,F6.3))
0040       113  FORMAT(1H1,3(X,' IRIS CORRECTION VECTORS (LABLED:CLASSIFIED)')
0041       114  FORMAT(1HO,14X,2(4X,'A1,'1',1A1,'3))
0042       115  FORMAT('1',42X,'DATA INTERFACE REPORT')
0043       116  FORMAT(1HO,3X,'TAPE  ',  =',2A1,57X,'TAPE NUMBER =',15,'0')
0044       117  FORMAT(1HO,5X,'RUN ID =',6A1,1X,6A1)
0045       118  FORMAT(1HO,5X,' DIGITIZE DATES')
0046       119  FORMAT(1H ,47X,'1',5X,'2',5X,'3',5X,'4')
0047       120  FORMAT(1H ,4X,'******  SEGMENT NUMBER =',4A1,13X,4(5A1,2X))
0048       121  FORMAT(1H ,'TRAI  I  SEGMENT NO. ',I1,' =',6A1)
0049       122  FORMAT(1H0,47X,'TYPE',1A1,' OUT LABEL/CLASSIFICATION')
0050       123  FORMAT(1HO,47X,'TYPE',1A1,' OUT LABEL/CLUSTER ASSIGNMENT')
```

```
0051    124     FORMAT(1H0,1X,'BIAS')
0052    125     FORMAT(' ',1X,'CLUSTER',2X,15(I3,3X))
0053      126   FORMAT(1H+,74X,'PCC',4X,'TH',6X,'UN',3X,'CORR?')
0054      127   FORMAT(1H+,74X,'PCC',5X,'CA',6X,'UN',3X,'CORR?')
0055      128   FORMAT(1H1,41X,'BIAS CORRECTION CLUSTER REPORT')
0056    129     FORMAT(1H1,10X)
0057    130     FORMAT(1H0,3X,'PCC1  = ',F6.1,3X,' PCC2  = ',F6.1,6X
     *  'PCC3  = ',F6.1,6X,' PCC12 = ',F6.1,6X,
     2' DW    = ',F6.1,6X,' DS    = ',F6.1)
0058    132     FORMAT(1H0,10X)
0059    133     FORMAT(1H+,70X,'CLUSTERED ')
0060    134     FORMAT(' ',15X,15(A1,'I',A1,I3,2X))
0061    135     FORMAT(1H0,22X,'TOTAL = ',I6)
0062      145   FORMAT(1H ,5X,'CORRECTED     % ',6(2X,I6),26X,I6)
0063      146   FORMAT(1H ,5X,'VAR AND DF    % ',6(2X,F6.2),26X,F6.2)
0064      147   FORMAT(1H ,5X,'RANDOM SAMPLE  ',6(2X,F6.2),26X,F6.2)
0065    148     FORMAT(1H+,102X,'VARIANCE IS UNDEFINED')
0066    150     FORMAT(1H0,45X,'INTERMEDIATE VALUES FOR DW AND DS')
0067    151     FORMAT(1H ,36X,' W',58X,'DS'/6X,'P=',4X,8I6,6X,'P=',4X,8I6)
0068    152     FORMAT(1H ,2X,'T(P)*(P)  ',8F6.3,' V(P)*S(P)  ',8F6.3)
0069    153     FORMAT(1H ,29X,'P(P,Q)*(P)*(Q)',46X,'U(P,Q)*S(P)*S(Q)')
0070    154     FORMAT(1H ,2X,'SET',I4,4X,8F6.2,12X,8F6.2)
0071    155     FORMAT(1H ,15X,' OFFSET CONSTANT FOR DW =',F10.5,
     1 25X,' OFFSET CONSTANT FOR DS =',F10.5)
0072            GO TO (1,2),PASS
0073        1   WRITE(6,115)
0074            DO 3 I=1,4
0075    3       SEQN(I)=IBUF(147+I)
0076            WRITE(6,116) (IBUF(I),I=3,26),TAPN
0077            WRITE(6,117) (IBUF(I),I=709,714),(IBUF(J),J=715,720)
0078            WRITE(6,118)
0079            WRITE(6,119)
0080            WRITE(6,120) (SEQN(I),I=1,4),(IBUF(J),J=154,158),
     2(IBUF(K),K=159,163),(IBUF(L),L=164,168),(IBUF(M),M=169,173)
0081            RETURN
0082    2       IF(.NOT.BUF,EQ.1) GO TO 202
0083            WRITE(6,129)
0084            CNT=0
0085            DO 97 JJ=1,2
0086            K=1
0087            NLI=0
0088            WRITE(6,122) CNT
0089            WRITE(6,124)
0090            WRITE(6,125) (CL(I),I=1,19)
0091            DO 98 J=1,11
0092            L=6
0093            DO 11 I=1,120
0094            PMAT(I)=BLANK
0095    10      CONTINUE
0096            DO 12 I=1,19
0097            IF(CL(I),EQ,'1') GO TO 6
0098            IF(TYPE(K),EQ,'1'.OR.TYPE(K),EQ,'2') GO TO 7
0099            GO TO 5
0100    6       IF(TYPE(K),EQ,'1'.OR.TYPE(K),EQ,'3') GO TO 7
0101            GO TO 5
0102    7       PMAT(L)=PLED(K)
```

```
0103            PARAM(I+1)=CLASX
0104            PARAM(I+2)=CLASFY(2*K)
0105          . IF(CLASFY(2*K-1),EQ,' ') GO TO 5
0106            PARAM(I+3)=CLASFY(2*K-1)
0107            PARAM(I+4)=CLASFY(2*K)
0108       5    CONTINUE
0109            K=K+1
0110            I=I+5
0111       12   CONTINUE
0112            WRITE(6,99) (PARAM(I),I=6,125)
0113       99   FORMAT(1HC,2X,13,2X,115A1)
0114            NLINE=NLINE+10
0115       98   CONTINUE
0116            NLINE=0
0117    97      CONTINUE
0118            WRITE(6,132)
0119            WRITE(6,101)
0120            WRITE(6,102) (PARAFL(I),I=1,10)
0121            WRITE(6,103) (PAR2P(I),I=1,10)
0122            WRITE(6,104) (PARVP(I),I=1,10)
0123            WRITE(6,105) (PAR2F(I),I=1,10)
0124            WRITE(6,106) (PARAR(I),I=1,10)
0125            WRITE(6,107) (PARANS(I),I=1,10)
0126            WRITE(6,132)
0127            WRITE(6,101)
0128            WRITE(6,102) (PARAFL(I),I=11,20)
0129            WRITE(6,103) (PAR2P(I),I=11,20)
0130            WRITE(6,104) (PARVP(I),I=11,20)
0131            WRITE(6,105) (PAR2F(I),I=11,20)
0132            WRITE(6,106) (PARAR(I),I=11,20)
0133            WRITE(6,107) (PARANS(I),I=11,20)
0134            WRITE(6,132)
0135            WRITE(6,101)
0136            WRITE(6,102) (PARAFL(I),I=21,26)
0137            WRITE(6,124)
0138            WRITE(6,103) (PAR2P(I),I=21,30)
0139            WRITE(6,104) (PARVP(I),I=21,30)
0140            WRITE(6,145) (PAR2R(I),I=21,26),BAR2R(30)
0141            WRITE(6,146) (PARAR(I),I=21,26),BARAR(30)
0142            IF(BARAR(30).EQ.0.0.AND.BAR2P(30).GT.0)WRITE(6,148)
0143            WRITE(6,147) (PARANS(I),I=21,26),BARANS(30)
0144            WRITE(6,132)
0145            WRITE(6,130) PCC1,PCC2,PCCG1,PCCG2,CLAD(1),CLAD(2)
0146            WRITE(6,135)
0147            WRITE(6,150)
0148            WRITE(6,155) CMZ STIF,CCFAS(2)
0149            WRITE(6,151) (I,I=1,6),(J,J=1,6)
0150            WRITE(6,152) ((BMTCH(I,J),J=1,6),I=1,2)
0151            WRITE(6,153)
0152            DO 180 K=1,6
0153            WRITE(6,154) K,((HARMON(L,K,M),M=1,6),L=1,2)
0154   180      CONTINUE
0155            WRITE(6,132)
0156            NC=1
0157            IF(NCATS.LE.13) NC=NCATS
0158            NYE=1
```

```
FORTRAN IV PLUS V02-51          11:09:33    20-JUL-78          PAGE 4
DIAPRT,FTN       /TR:3 "CKS/AR

0159            WRITE(6,108)
0160            WRITE(6,109) (I,I=1,NC)
0161            WRITE(6,110) (BACLAB(I),I=1,NC)
0162            WRITE(6,111)
0163            DO 96 J=1,BACATS
0164            WRITE(6,112) BCLABL(J),(BACAIM(J,K),K=1,NC)
0165    96      CONTINUE
0166            IF(BACATS.LE.13) GO TO 196
0167            NC=BACATS
0168            WRITE(6,132)
0169            ITE=13
0170            WRITE(6,109) (I,I=14,NC)
0171            WRITE(6,110) (BACLAB(I),I=14,NC)
0172            WRITE(6,111)
0173            DO 194 J=1,BACATS
0174            WRITE(6,112) BCLABL(J),(BACAIM(J,K),K=14,BACATS)
0175    196     CONTINUE
0176            NC=7
0177            IF(BACATS.LE.13) NC=BACATS
0178            WRITE(6,113)
0179            DO 50 K=1,BACATS
0180            WRITE(6,134) (BCVAR(K),BACLAB(LL),BCVA(K,LL),LL=1,NC)
0181    50      CONTINUE
0182            WRITE(6,132)
0183            IF(BACATS.LE.13) GO TO 161
0184            ITE=BACATS
0185            DO 160 K=1,BACATS
0186            WRITE(6,134) (BCVAR(K),BACLAB(LL),BCVA(K,LL),LL=14,NC)
0187    160     CONTINUE
0188    161     CONTINUE
0189            WRITE(6,135) DATVT
0190    202       IF(CLUMC.EQ.1) RETURN
0191            WRITE(6,129)
0192            DATE=0
0193            DO 207 JJ=1,2
0194            K=1
0195            NLIN=0
0196            WRITE(6,123) DAY
0197            WRITE(6,124)
0198            WRITE(6,125) (CLU(I),I=1,15)
0199            DO 209 J=1,11
0200            L=6
0201            DO 210 I=1,120
0202            PVATR(I)=LANK
0203    210     CONTINUE
0204            DO 214 I=1,15
0205            IF(CLU.EQ.11) GO TO 206
0206            IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'2') GO TO 207
0207            GO TO 205
0208    206       IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'3') GO TO 207
0209            GO TO 206
0210    207     PVATR(L)=LABLD(K)
0211            PVATR(L+1)=SLAS
0212            PVATR(L+2)=CLU2(I,K)
0213            IF(CLU2(2*K-1).EQ.' ') GO TO 5
0214            PVATR(L+3)=CLU2(2*K-1)
```

```
0215            RWATR(L+3)=CLDPT(2*K)
0216      205   CONTINUE
0217            K=K+1
0218            L=L+6
0219      212   CONTINUE
0220            WRITE(6,99) NLINE,(RWATR(I),I=6,120)
0221            NLINE=NLINE+1
0222      298   CONTINUE
0223            GO TO 2
0224      297   CONTINUE
0225            WRITE(6,128)
0226            WRITE(6,101)
0227            WRITE(6,102) (PCLAPL(I),I=1,10)
0228            WRITE(6,103) (PCLAP(I),I=1,10)
0229            WRITE(6,104) (PCLWC(I),I=1,10)
0230            WRITE(6,105) (PCLCP(I),I=1,10)
0231            WRITE(6,106) (PCLAR2(I),I=1,10)
0232            WRITE(6,107) (PCLAWS(I),I=1,10)
0233            WRITE(6,101)
0234            WRITE(6,102) (PCLAPL(I),I=11,20)
0235            WRITE(6,103) (PCLAP(I),I=11,20)
0236            WRITE(6,104) (PCLWC(I),I=11,20)
0237            WRITE(6,105) (PCLCP(I),I=11,20)
0238            WRITE(6,106) (PCLAR2(I),I=11,20)
0239            WRITE(6,107) (PCLAWS(I),I=11,20)
0240            WRITE(6,101)
0241            WRITE(6,102) (PCLAPL(I),I=21,26)
0242            WRITE(6,127)
0243            WRITE(6,103) (PCLAP(I),I=21,30)
0244            WRITE(6,104) (PCLWC(I),I=21,30)
0245            WRITE(6,148) (PCLCP(I),I=21,26),BUCOR(30)
0246            WRITE(6,149) (PCLAR(I),I=21,26),BUVAR(30)
0247            IF(BUVAR(30).EQ.0.0.AND.BLP2P(30).GT.0)WRITE(6,148)
0248            WRITE(6,147) (PCLAWS(I),I=21,26),BUWAWS(30)
0249            WRITE(6,139)
0250            WRITE(6,137) PCL1,PCL2,PCL31,PCL32,CLUD(1),CLUD(2)
0251            WRITE(6,138)
0252            WRITE(6,154)
0253            WRITE(6,155) CCWS(1),CCWS(2)
0254            WRITE(6,151) (I,I=1,8),(J,J=1,8)
0255            WRITE(6,152) ((B TO(I,J),J=1,8),I=1,2)
0256            WRITE(6,153)
0257            DO 190 K=1,6
0258            WRITE(6,154) K,((BUWAWS(L,K,M),M=1,8),L=1,2)
0259      190   CONTINUE
0260            WRITE(6,132)
0261            WRITE(6,132)
0262            NC=8
0263            IF(NCATS.LE.13) NC=NCATS
0264            KK=1
0265            WRITE(6,109)
0266            WRITE(6,110) (I,I=1,NC)
0267            WRITE(6,111) (PCLA (I),I=1,NC)
0268            WRITE(6,111)
0269            DO 211 J=1,PLCATS
0270            WRITE(6,112) PCLAPL(J),(BUCAIR(J,K),K=1,NC)
```

```
0271        20    CONTINUE
0272              IF(NCATS.LE.13) GO TO 421
0273              NC=13
0274              WRITE(6,132)
0275              NT=NC+13
0276              WRITE(6,108) (I,I=14,NC)
0277              WRITE(6,110) (BUCLAB(I),I=14,NC)
0278              WRITE(6,111)
0279              DO 42[illegible] J=1,BUCATS
0280              WRITE (6,112) TLOAPL(J),(BUCAIN(J,K),K=14,BUCATS)
0281     420      CONTINUE
0282     421      CONTINUE
0283              NC=13
0284              IF(NCATS.LE.13) NC=BUCATS
0285              WRITE(6,113)
0286              DO 25[illegible] K=1,BUCATS
0287              WRITE(6,134) (BUCLAB(K),BUCLAB(LL),BCVU(K,LL),LL=1,NC)
0288     250      CONTINUE
0289              WRITE(6,139)
0290              IF(NCATS.LE.13) GO TO 450
0291              NC=NCATS
0292              DO 45[illegible] K=1,BUCATS
0293              WRITE(6,134) (BUCLAB(K),BUCLAB(LL),BCVU(K,LL),LL=14,NC)
0294     450      CONTINUE
0295              WRITE(6,135) BUDGET
0296              RETURN
0297              END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	017710	3044	RW,I,CON,LCL
2	$PDATA	000006	3	RW,D,CON,LCL
3	$IDATA	000504	674	RW,D,CON,LCL
4	$VARS	000455	155	RW,D,CON,LCL
5	$TEMPS	000006	3	RW,D,CON,LCL
6	CBIAS	026742	5873	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLODUM	000442	290	RW,D,OVR,GBL
9	SEG	000004	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BIAPRT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
AP2P	I*2	6-016150	AUNCA	I*2	6-016066	BACATS	I*2	6-016032	BATPT	I*2	6-023352	BLANK	L*1	4-000170
BUCATS	I*2	6-014076	BUTPT	I*2	6-023354	CLADUM	I*2	7-000000	CLUDUM	I*2	7-000002	CNT	I*2	4-000452
DP	I*2	6-014200	DT	I*2	6-014202	I	I*2	4-000440	J	I*2	4-000442	JJ	I*2	4-000454
K	I*2	4-000444	L	I*2	4-000446	LL	I*2	4-000454	M	I*2	4-000450	N	I*2	4-000460
NC	I*2	4-000462	NLARG	I*2	6-026706	NLINE	I*2	4-000456	NT	I*2	4-000436	ONE	I*2	4-000430
PASS	I*2	6-016076	PCCG1	R*4	6-016100	PCCG2	R*4	6-016104	PCCT	R*4	6-016070	PCC2	R*4	6-016074
PCLG1	R*4	6-016120	PCLG2	R*4	6-016124	PCL1	R*4	6-016110	PCL2	R*4	6-016114	SLASH	I*2	4-000432
TAPNO	I*2	6-016146	TC	I*2	6-014206	TWO	I*2	4-000434						

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-014150	000030	12	(3,2)
BACAIN	R*4	6-020050	005220	1342	(25,26)
BACLAB	L*1	6-016134	000032	26	(26)
BACTR	I*2	6-020050	000074	34	(37)
BALABL	L*1	6-016134	000032	26	(26)
BAPTR	I*2	6-020050	000074	34	(37)
BARAYS	R*4	6-016152	000170	60	(30)
BARCAN	R*4	6-022176	001040	256	(2,8,8)
BATCAN	R*4	6-022154	000110	32	(2,8)
BAUCTR	I*2	6-020050	000074	34	(37)
BAUAR	R*4	6-016152	000170	60	(30)
BCAIN	I*2	6-020050	005510	676	(25,26)
BCAIN	I*2	6-020050	005510	676	(25,26)
BUCAIN	R*4	6-020050	005220	1342	(25,26)
BUCLAB	L*1	6-016134	000032	26	(26)
BUCTR	I*2	6-020050	000074	34	(37)

B[illegible]	L*1	6-[illegible]	[illegible]	13	(26)
B[illegible]	I*2	6-[illegible]	[illegible]	30	(30)
BUFAD	R*4	6-[illegible]	000170	60	(30)
[illegible]	R*4	6-[illegible]	[illegible]	256	(2,8,8)
[illegible]	I*4	6-[illegible]	[illegible]	2	(2,8)
[illegible]	I*2	6-[illegible]	[illegible]	30	(30)
BUFAR	R*4	6-[illegible]	000170	60	(30)
CCOMS	R*4	6-[illegible]	[illegible]	4	(2)
CLAD	R*4	6-[illegible]	[illegible]	4	(2)
CLASFY	L*1	6-[illegible]	000542	209	(418)
CLOAT	L*1	6-[illegible]	000642	209	(418)
CLOD	R*4	6-[illegible]	[illegible]	4	(2)
CPL	I*2	4-[illegible]	[illegible]	19	(19)
GRAINS	L*1	6-[illegible]	[illegible]	3	(25)
IRUF	L*1	F-[illegible]	[illegible]	1	(1)
LPLED	L*1	6-[illegible]	000321	104	(209)
PP	L*1	4-[illegible]	000170	60	(120)
PPATR	L*1	4-[illegible]	000170	60	(120)
RCOMS	R*4	6-[illegible]	[illegible]	256	(2,8,8)
SPSWE	L*1	9-[illegible]	[illegible]	2	(4)
TCOMS	R*4	6-[illegible]	[illegible]	32	(2,8)
TYPE	L*1	6-[illegible]	000321	104	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000054	2	1-000630	3	**	5	1-001344	6	1-001216
7	1-001254	11	**	12	**	20	**	50	**
96	**	97	**	98	**	99'	3-002454	100'	3-000000
101'	3-000054	102'	3-000074	103'	3-000112	104'	3-000154	105'	3-000206
106'	3-000244	107'	3-000304	108'	3-000344	109'	3-000376	110'	3-000424
111'	3-000460	112'	3-000502	113'	3-000524	114'	**	115'	3-000610
116'	3-000644	117'	3-000720	118'	3-000752	119'	3-001002	120'	3-001026
121'	**	122'	3-001104	123'	3-001134	124'	3-001230	125'	3-001244
126'	3-001270	127'	3-001326	128'	3-001364	129'	3-001432	130'	3-001442
132'	3-001600	133'	**	134'	3-001606	135'	3-001632	145'	3-001654
146'	3-001716	147'	3-001762	148'	3-002026	150'	3-002062	151'	3-002132
152'	3-002200	153'	3-002250	154'	3-002324	155'	3-002356	160	**
161	1-006236	189	**	190	**	196	1-005552	202	1-006274
205	1-007712	206	1-005660	207	1-006716	210	**	212	**
250	**	297	**	298	**	420	**	421	1-013206
45"	1-013525								

TOTAL SPACE ALLOCATED = 044732 9965

NO FPP INSTRUCTIONS GENERATED

BIAPRT,LP1=BIAPRT

```
F2FTRAN IV-PLUS V02-00          10:19:15C     24-JUL-78          PAGE 1
C:\REC.FTN       /FF;R:80COLS/NOWARN
C
C     TYPE OF REC
      IF (INIT.EQ.1) GO TO 1
      BYTE BSPV(41B),BCLV(41B)
      INTEGER*1 TYPE(259),LBLEC(209)
      INTEGER*1  BACLAB(26),BUCLAB(26),BULABL(26),GRAINS(26)
      REAL*4 T,PCC1,PCC2,PCC3,PCL1,PCL2,PCLG1,PCLG2
      REAL BCLAB,BCLASS,TYPE,LABEL,BCCATS,BUCATS
      REAL APPP,BACLAB,BUCLAB
      DIMENSION C(2),C1(2),TCATS(2,8),CCATS(2),RCATS(2,8,8)
      DIMENSION BACATS(2,8),BCTCAT(2,8),BARCAT(2,8,8),BURCAT(2,8,8)
      DIMENSION ICL(2,8),CVL(26,26)
      DIMENSION BACAT(30),BUCAT(30)
      COMMON/CBIGS/A,B,  PPP,BACCP,BUCCP,BACAP,BUCAP,BACAIN,BUCAIN,
     *           T,C,  AT,BARATS,BURATS,A,B,CM,TC,BUCATS,BUCLAB,
     *           C1,C2,CLABL,CLASSV,TYPE,LABEL,BACATS,BACLAB,AUCC2,
     *           PCC1,PCC2,PCCC2,PCCC3,PCL1,PCLP,PCLG1,PCLG2,APPP,
     *           ICL,A1,,RAT2T,BLT2T,CLAD,CLLD,TCATS,RCATS,CCATS,
     *           BACAT,BUCAT,BARCAT,BURCAT,LABEL,GRAINS
      COMMON/DUMMY/CVL,BCLASS
      COMMON/CLCOM/TL,BT
      REAL A(3,2)
      DIMENSION BACAT(30),BUCCP(30),BACAP(30),BUCAP(30)
      REAL BACAIN(26,2),  CAIN(26,26),BAVAR(30),BUVAR(30)
      REAL PARAMS(30),  CATS(30)
      IC=1
      CLA=0
      DO 5 I=1,26
      GRAINS(I)=' '
    5 DO 6 I=1,2
      READ(1,101) CVL(I)
  101 FORMAT(F10.2)
      READ(1,102) (TCATS(I,J),J=1,8)
      DO 9 J=1,8
      READ(1,102) (RCATS(I,J,K),K=1,8)
    9 CONTINUE
  102 FORMAT(8F10.6)
      TYPE 56
   56 FORMAT(1H ,' I P T CHRBZ 1')
      ACCEPT 55,IPTS
      READ(IPTS,55)GRAINS
   55 FORMAT(26A1)
      DO 60 I=1,26
      IF(GRAINS(I).EQ.' ') GOTO 65
   60 CLA=CLA+1
   65 IF(CLA.GT.0) NATL=1
      GRAINS(1)='S'
      GRAINS(2)='T'
      GRAINS(3)='-'
      CLA=0
      RETURN
      END
```

PROGRAM SECTIONS

NUMBER	NAME	TYPE		ATTRIBUTES
1	BLOCK1	OCT	194	RW,I,GBL,LCL
3	PIDATA	OCT	11	RW,D,GBL,LCL
4	.BANK	OCT	1	RW,D,GBL,LCL
5	DATAS	OCT	5823	RW,D,REL,GBL
7	.BANK	OCT	2	RW,D,REL,GBL
8	BLOCK2	OCT	279	RW,D,REL,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CONFIG		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
APLPR	I*2	6-016036	ARG	I*2	6-016066	BACATS	I*2	6-016032	BATST	I*2	6-023352	BUCATS	I*2	6-014236
BLTST	I*2	6-023454	CLASS4	I*4	7-000000	CLUDUM	I*2	7-000002	D0	I*2	6-014200	DU	I*2	6-014202
I	I*2	4-000002	IC	I*2	4-000000	J	I*2	4-000004	K	I*2	4-000006	NLARG	I*2	6-026706
PCCG1	R*4	6-016104	PCLW2	R*4	6-016104	PCC1	R*4	6-016070	PCC2	R*4	6-016074	PCLG1	R*4	6-016120
PCLG2	R*4	6-016124	PCL1	R*4	6-016110	PCL2	R*4	6-016114	TC	I*2	6-014204			

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-014150	000030	12	(3,2)
BACATS	R*4	6-[illegible]	000220	144	(26,26)
BACLAS	L*1	6-014[illegible]	000032	13	(26)
BADPR	I*2	6-[illegible]	000074	30	(30)
BALABL	L*1	6-014[illegible]	000032	13	(26)
BAPPR	I*2	6-[illegible]	000074	30	(30)
BARANS	R*4	6-013[illegible]	000170	60	(30)
BARCDS	R*4	6-023[illegible]	001000	256	(2,8,8)
BATCDS	R*4	6-024[illegible]	000010	8	(2,8)
BAVNOD	I*2	6-[illegible]	000074	30	(30)
BAVAR	R*4	6-013[illegible]	000170	60	(30)
BCVAR	I*2	6-[illegible]	000544	676	(26,26)
BCVU	I*2	6-[illegible]	000544	676	(26,26)
BUCATS	R*4	6-[illegible]	000220	144	(26,26)
BUCLAS	L*1	6-014[illegible]	000032	13	(26)
BUDPR	I*2	6-[illegible]	000074	30	(30)
BULABL	L*1	6-014[illegible]	000032	13	(26)
BUPPR	I*2	6-[illegible]	000074	30	(30)
BURANS	R*4	6-014[illegible]	000170	60	(30)
BURCDS	R*4	6-025[illegible]	001000	256	(2,8,8)
BUTCDS	R*4	6-[illegible]	000010	8	(2,8)
BUVNOD	I*2	6-[illegible]	000074	30	(30)

COMMON,FTN /TR:1 /CROSS/...

BLVAR	R*1	5-[illegible]	000170	41	(33)
CCOS	R*4	5-[illegible]	[illegible]	4	(2)
CLAD	R*4	5-[illegible]	[illegible]	4	(2)
CLASFY	L*1	5-[illegible]	[illegible]52	2*9	(41F)
CLOST	L*1	5-[illegible]	[illegible]542	2*9	(42E)
CLP	R*4	5-[illegible]	[illegible]4F	4	(2)
COATS	L*1	5-[illegible]	[illegible]%	13	(26)
DELED	L*1	5-[illegible]	[illegible]321	1*2	(2*9)
ECO S	R*4	5-[illegible]	101[illegible]	25*	(2,5,5)
EQUIS	R*4	5-[illegible]	[illegible]1[illegible]	32	(2,8)
TYPE	L*1	5-[illegible]	[illegible]371	1*4	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
5	**	9	**	10	**	50	**	55'	3-000012
60	**	65	1-[illegible]0510	100'	3-000004	101'	3-000000		

TOTAL SPACE ALLOCATED = 030426 4283

NO FPP INSTRUCTIONS GENERATED

COMMAND,LPT=COMMAND

```
0153              IF(TRUF(1,N).EQ.CAT(JJ)) GØ TØ 270
0154     260      CØNTINUE
0155              NC=NC+1
0156              CAT(NC)=TRUF(1,N)
0157     270      CØNTINUE
                  C NØW SØRT THE ENTRIES BY DØT DISTANCE
0158     271      DØ 280 II=1,N
0159              T=9999
0160              DØ 275 JJ=1,N
0161              IF(T.LT.DRUF(JJ)) GØ TØ 275
0162              PTAR(II)=JJ
0163              T=DRUF(JJ)
0164     275      CØNTINUE
0165              JK=PTAR(II)
0166              DRUF(JK)=9999
0167     280      CØNTINUE
0168     44       CØNTINUE
0169       105    FØRMAT(I5)
0170              XL2=L2DIST(J)
0171              XL2=XL2/100.
0172     210      FØRMAT(I2)
0173              WRITE(3,201)
0174     201      FØRMAT('0',70X,'BRIGHTNESS  GREEN NUMBER')
0175     202      FØRMAT(I4)
0176              WRITE(3,203) (CLASSE(L,J),L=1,6),A1B(J),A1G(J)
0177     203      FØRMAT(' ',16X,'CLUSTER NAME: ',2X,6A1,18X,'ACQUISITIØN 1',6
                2I4,8X,I3)
0178              WRITE(3,211) (SCLASS(L,J),L=1,6),A2B(J),A2G(J)
0179     211      FØRMAT(' ',16X,'LABELING DØT: ',2X,6A1,18X,'ACQUISITIØN 2',6
                2I4,8X,I3)
0180              WRITE(3,212) XL2,A3B(J),A3G(J)
0181     212      FØRMAT(' ',16X,'L2 DISTANCE: ',2X,F6.2,18X,'ACQUISITIØN 3',
                2I4,8X,I3)
0182              WRITE(3,213) CAT,A4B(J),A4G(J)
0183     213      FØRMAT(' ',16X,'CATAGØRIES:  ',2X,8(X,A1),8X,'ACQUISITIØN 4
                2I4,8X,I3)
0184              WRITE(3,99)
                  C
                  C   NØW ØUTPUT DØTS IF ANY
0185              IF(N.EQ.0) GØ TØ 106
0186              WRITE(3,214)
0187     214      FØRMAT(' ',10X,'DØTS    DISTANCE    DØTS    DISTANCE    D
                2'    DISTANCE    DØTS    DISTANCE    DØTS    DISTANCE')
0188              WRITE(3,99)
0189              DØ 216 II=1,N,5
0190              FF=II+4
0191              LIM=5
0192              IF(FF.LE.N) GØ TØ 220
0193              FF=N
0194              LIM=N-II+1
0195     220      LL=1
0196              DØ 217 JJ=II,FF
0197              NDX=PTAR(JJ)
0198              DØ 218 KK=1,5
0199              DD(KK,LL)=TRUF(KK,NDX)
0200     218      CØNTINUE
```

```
0201                DIS(LL)=DSBUF(NDX)
0202                LIS(LL)=DIS(LL)/1[illegible]0.
0203                LL=LL+1
0204      217       CONTINUE
0205                WRITE(3,215) ((DI(KK,JJ),KK=1,5),DIS(JJ),JJ=1,LIM)
0206      216       CONTINUE
0207      215       FORMAT(' ',9X,5(A1,1X,3A1,1X,A1,3X),F6.2,5X))
                C
                C TEST FOR END OF DATA
                C
0208      104       IF(J.EQ.ALSETS) GO TO 4
0209                J=J+1
0210              IF(K.EQ.MAXC) GO TO 15
0211                K=K+1
0212                GO TO 2
                C       FINISHED 1 PAGE, GO TOP OF FORM AND HEADER
0213      15        WRITE(3,98)
0214                WRITE(3,100)
0215              MAXC=5
0216                GO TO 1
                C
0217       4        CONTINUE
0218                WRITE(3,50)
0219       50       FORMAT('0',10X,'CLUSTERING CHANNEL LIST:')
0220                WRITE(3,51)  (CT(I),I=1,CHMAX)
0221       51       FORMAT(' ',10X,16(1X,I2))
0222                RETURN
0223              END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	005076	1311	RW,I,CON,LCL
3	$IDATA	001214	326	RW,D,CON,LCL
4	$VARS	011464	2458	RW,D,CON,LCL
5	$TEMPS	000012	5	RW,D,CON,LCL
6	CBIAS	005122	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLCOM	000642	209	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLURES		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALSETS	I*2	4-011376	BACATS	I*2	6-003314	BATPT	I*2	6-003346	BLANK	I*2	4-011376	BUCATS	I*2	6-001556
BUTPT	I*2	6-003370	CMPMX	I*2	4-011406	CLABUM	I*2	7-000000	CLUBUM	I*2	7-000002	D	I*2	4-011442
DFLD	I*2	4-011416	DP	I*2	6-001550	DRPT	I*2	4-011410	DU	I*2	6-001552	EF	I*2	4-011454
FILE	I*2	F-000006*	I	I*2	4-011402	II	I*2	4-011434	IJ	I*2	4-011432	J	I*2	4-011404
JJ	I*2	4-011426	JK	I*2	4-011450	K	I*2	4-011420	KK	I*2	4-011442	L	I*2	4-011452
LFLD	I*2	4-011414	LIM	I*2	4-011456	LL	I*2	4-011440	MAXC	I*2	4-011430	N	I*2	4-011440
NC	I*2	4-011444	NDX	I*2	4-011436	NUM	I*2	4-011424	N1	L*1	4-003506	N2	L*1	4-006307
PCCG1	R*4	6-003336	PCCG2	R*4	6-003342	PCC1	R*4	6-003326	PCC2	R*4	6-003332	PCLG1	R*4	6-003356
PCLG2	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	PT	I*2	4-011422	R	I*2	F-000004*
RJ	I*2	4-011412	SETSR	I*2	4-011403	T	I*2	4-011446	TC	I*2	6-001554	XB	R*4	4-000004
XG	R*4	4-000010	XL2	R*4	4-000000									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
A1B	I*2	4-007400	000170	60	(60)
A1G	I*2	4-007210	000170	60	(60)
A2B	I*2	4-007760	000170	60	(60)
A2G	I*2	4-007570	000170	60	(60)
A3B	I*2	4-010340	000170	60	(60)
A3G	I*2	4-010150	000170	60	(60)
A4B	I*2	4-010720	000170	60	(60)
A4G	I*2	4-010530	000170	60	(60)
BACAIN	R*4	6-000220	001400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACPR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPMP	I*2	6-000030	000030	12	(12)
BARANS	R*4	6-001360	000060	24	(12)

BAUNCR	I*2	6-000361	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003356	000200	64	(8,8)
BCVB	I*2	6-003564	000200	64	(8,8)
BUCAR	R*4	6-005620	000470	194	(5,5)
BUFLAG	L*1	6-001540	000010	4	(2)
BUCAR	I*2	6-000170	000030	12	(12)
BVLABL	L*1	6-001570	000010	4	(2)
BVPOP	I*2	6-000630	000130	52	(13)
BURANS	R*4	6-001640	000060	24	(12)
BVUNCR	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001720	000060	24	(12)
CAT	L*1	4-000040	000010	4	(2)
CCRAS	R*4	6-005112	000010	4	(2)
CD.AME	I*2	4-004724	000170	60	(60)
CLAR	R*4	6-003772	000010	4	(2)
CLASSY	L*1	6-001610	000642	209	(418)
CLERT	L*1	6-000010	000652	209	(418)
CLOD	R*4	6-004012	000010	4	(2)
CLARE	L*1	4-005650	000553	150	(6,60)
CT	I*2	4-006570	000140	16	(16)
CRUF	I*2	4-007114	000074	30	(30)
DD	L*1	4-004536	000021	12	(5,5)
DIS	R*4	4-005014	000024	10	(5)
DTRUF	L*1	4-004370	004716	*259	(2515)
DSBUF	I*2	4-011300	000074	30	(30)
IBUF	L*1	F-000002*	000001	0	(1)
LELGD	L*1	6-002773	000321	104	(209)
L2LIST	I*2	4-011110	000170	60	(60)
PTAS	I*2	4-004630	000074	30	(30)
RCCNS	R*4	6-004112	001003	256	(2,8,8)
SCLASS	L*1	4-000620	000550	150	(6,50)
TRUF	L*1	4-004310	000226	75	(5,30)
TCCNS	R*4	6-004312	000110	72	(2,8)
TYPE	L*1	6-002452	000321	104	(209)

#

#

LABELS

#

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-002452	2	1-002466	4	1-002477*	10	1-000057*	12	1-000732
15	1-004710	19	1-000434	20	**	21	**	3*	1-001474
39	**	40	**	41	**	42	**	4*	1-002164
44	1-003432	45	**	46	1-002414	47	1-002266	5**	3-001104
51*	3-001146	98*	3-000004	99*	3-000040	100*	3-000062	102*	3-000000
103*	3-000306	105*	3-000312	106	1-004644	110*	3-000046	200*	3-001134
201*	3-000316	202*	3-000356	203*	3-000360	204*	3-000300	21**	**
211*	3-000444	212**	3-000626	213*	3-000610	214*	3-000674	215*	3-001756
216	**	217	**	218	**	220	1-004256	240	1-002770
250	**	270	**	270	1-003200	271	1-003222	275	1-003340
280	**	1000*	3-000130	1001*	3-000170	1002*	3-000210	1*****	3-000240

#

#

FUNCTIONS AND SUBROUTINES REFERENCED

#

CLURES,FTN /TR:BLØCKS/WP
CDRED
@
@
TØTAL SPACE ALLØCATED = 026000 5632
@
CLURES,LP1=CLURES

```
STDATA.FTN       /TRIPLOCKS/R
0001             SUBROUTINE STDATA(IBUF,R,FILE,DFLG)
0002             IMPLICIT INTEGER(A-Z)
0003             INCLUDE 'COMMON1.DAT'
0004  *          BYTE CLASFY(418),TYPE(418)
0005  *          LOGICAL*1 TYPE(209),LABLED(209)
0006  *          LOGICAL*1 BALABL(8),BACLAB(8),BUCLAB(8),BULABL(8)
0007  *          REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
0008  *          REAL CLAD,CLUD,TCONS,RCONS,CCONS
0009  *          DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,8,8)
0010  *          DIMENSION BCVA(8,8),BCVU(8,8)
0011  *          DIMENSION PAYBP(12),BUYBP(12)
0012  *          COMMON/COMAS/BAPAP,BUPBP,BALNCB,BUUNCB,BACBR,BUCBR,BACAIN,BUCAIN,
     *  *                  BAVAR,BUVAR,BAPANS,BUPANS,A,DZ,DN,TC,BUCATS,BUPLAB,
     *  *                  BULABL,BALBUL,CLASFY,TYPE,LABLED,BACATS,BUCLAB,
     *  *                  PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
     *  *                  BCVA,BCVU,PATBT,BUTBT,CLAD,CLUD,TCONS,RCONS,CCONS
0013  *          COMMON/DUMMY/CLADUM,CLUDUM
0014  *          COMMON/CLCOM/CLDAT
0015  *          REAL A(3,2)
0016  *          DIMENSION BAUNCB(12),BUUNCB(12),BACBR(12),BUCBR(12)
0017  *          REAL BACAIN(8,8),BUCAIN(8,8),BAVAR(12),BUVAR(12),BAPANS(12),BUPANS(12)
0018             LOGICAL*1 IBUF(1),PEN(132),PP(132),PN(132)
0019             LOGICAL*1 PTL(132),PK(2112)
0020             LOGICAL*1 BLANK
0021             LOGICAL*1 PF(132)
0022             REAL T1
0023             COMMON/PCNT/LINE,PAGE
0024             DATA BLANK/1H /
0025             IF(DFLG.EQ.0)  GO TO 12
0026             CALL STDMP(IBUF)
0027             GO TO 13
0028    12       CONTINUE
0029    13       CONTINUE
0030             NMAX=110
0031             DPAP=0
0032             K=1
0033             I=1
0034             PI=11
0035             DECODE(3,99,IBUF(5))  ALSETS
0036             IF(ALSETS.EQ.0)  RETURN
0037    99       FORMAT(I3)
0038             DECODE(2,98,IBUF(10))  NCH
0039    98       FORMAT(I2)
0040             LINE=66
0041             CALL BNT
0042             IF(IBUF(2).EQ.'S')  LINE=LINE+2
0043             NLINS=59-LINE
0044             LSAV=LINE
0045             XL=NCH*6
0046             IF(DFLG.EQ.1) NLINS=NLINS-7
0047             LSETS=NLINS/XL
0048             IF(LSETS.GT.0) WRITE(3,100)
0049      100    FORMAT(' ')
0050             IF(LSETS.GT.0)  GO TO 8
0051             LINE=66
0052             CALL BNT
```

```
STDATA,FTN      /TRIBLØCKS/WP
0053            LSETS=60/XL
0054     8      CØNTINUE
0055            IF(IBUF(2).EQ.'F')  GØ TØ 16
0056            WRITE(3,97)
0057       97   FØRMAT(' ',49X,'STATISTICS REPØRT')
0058       16   CØNTINUE
0059           ØSETS=5
0060            PSETS=0
0061           LFLD=11+9*NCH
0062           IF(IBUF(2).EQ.'F')  LFLD=18+9*NCH
0063     20    J=1
0064           RJ=12
0065           IF(K.EQ.1)  GØ TØ 6
0066           CALL CØRED(IBUF,R,FILE)
0067     4     DECØDE(2,98,IBUF(R))  SETSR
0068            IF(SETSR.EQ.0)  RETURN
0069     1      CØNTINUE
0070           IF(IBUF(2).EQ.'S')  GØ TØ 22
0071           CALL FNAMF(IBUF(RJ),PFN(PI))
0072     22    CALL PØP(IBUF(RJ),PP(PI),IBUF(2))
0073           CALL SNAMF(IBUF(RJ),PN(PI),IBUF(2))
0074           CALL MØTTL(PTL(PI))
0075            CALL MEAN(IBUF(RJ),PM(PI),NCH,IBUF(2))
0076           DECØDE(5,96,IBUF(RJ+6)) DUM1
0077     96     FØRMAT(I5)
0078           DPØP=DPØP+DUM1
0079           IF(K.EQ.ALSETS)  GØ TØ 10
0080           K=K+1
0081           IF(I.EQ.ØSETS)  GØ TØ 9
0082           I=I+1
0083           PI=PI+19
0084     2     IF(J.EQ.SETSR)  GØ TØ 20
0085           J=J+1
0086           RJ=RJ+LFLD
0087           GØ TØ 1
0088     9      K=K-1
0089     10    CØNTINUE
0090            PSETS=PSETS+1
0091            IF(PSETS.LE.LSETS)  GØ TØ 11
0092            PSETS=1
0093            IF(DFLG.EQ.1)  GØ TØ 14
0094            LINE=66
0095            CALL BNT
0096            LSETS=60/XL
0097            GØ TØ 11
0098     14     CØNTINUE
0099            LINE=66
0100            CALL KNT
0101     11     CØNTINUE
0102            WRITE(3,90)
0103     90     FØRMAT(1HØ,' ')
0104           IF(IBUF(2).EQ.'S')  GØ TØ 3
0105           WRITE(3,101)  (PFN(N),N=1,NMAX)
0106     101   FØRMAT(1H ,131A1)
0107           WRITE(3,102)  (PP(N),N=10,NMAX)
0108     102   FØRMAT(1H ,'CHANNEL ',122A1)
```

```
0109                WRITE(3,101)  (PN(N),N=1,NMAX)
0110                GØ TØ 7
0111      3          CØNTINUE
0112                WRITE(3,101)  (PN(N),N=1,NMAX)
0113                WRITE(3,102)  (PP(N),N=9,NMAX)
0114      7         WRITE(3,103)  (PTL(N),N=8,NMAX)
0115      103       FØRMAT(1H ,'NUMBER ',122A1)
0116                IN=6
0117                IM=NMAX
0118                DØ 5  II=1,NCH
0119                WRITE(3,104)  (II,(PM(N),N=IN,IM))
0120      104       FØRMAT(1H ,2X,I2,128A1)
0121                IN=IN+132
0122                IM=IM+132
0123      5         CØNTINUE
0124      6         CØNTINUE
0125                DØ  30 N=1,132
0126                PF(N)=BLANK
0127                 PFN(N)=BLANK
0128                PN(N)=BLANK
0129                PTL(N)=BLANK
0130                PP(N)=BLANK
0131                DØ  40  NN=1,NCH
0132                PM(N+(NN-1)*132)=BLANK
0133      40        CØNTINUE
0134      30        CØNTINUE
0135                IF(K.EQ.1)  GØ TØ 4
0136                IF(K.EQ.ALSETS)  GØ TØ 15
0137                 K=K+1
0138                I=1
0139                PI=11
0140                GØ TØ 2
0141      15         CØNTINUE
0142                 IF(LINE.EQ.0) LSAV=0
0143                 LINE=PSETS*XL+LSAV
0144                 CALL BNT
0145                BUPØP(10)=22932-DPØP
0146                BUUNCØ(10)=((22932.-DPØP)/22932.)*100
0147                 RETURN
0148                END
```

PRØGRAM SECTIØNS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CØDE1	003012	773	RW,I,CØN,LCL
2	$PDATA	000004	2	RW,D,CØN,LCL
3	$IDATA	000210	68	RW,D,CØN,LCL
4	$VARS	005410	1412	RW,D,CØN,LCL
5	$TEMPS	000006	3	RW,D,CØN,LCL
6	CBIAS	005122	1321	RW,D,ØVR,GBL
7	DUMMY	000004	2	RW,D,ØVR,GBL
8	CLCØM	000642	209	RW,D,ØVR,GBL
9	PCNT	000004	2	RW,D,ØVR,GBL

ENTRY PØINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
STDATA		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALSETS	I*2	4-005344	BACATS	I*2	6-003314	BATØT	I*2	6-003766	BLANK	L*1	4-005190	BUCATS	I*2	6-001556
BUTØT	I*2	6-003770	CLADUM	I*2	7-000000	CLUDUM	I*2	7-000002	DFLG	I*2	F-000010*	DØ	I*2	6-001550
DPØP	I*2	4-005334	DU	I*2	6-001552	DUM1	I*2	4-005374	FILE	I*2	F-000006*	I	I*2	4-005340
II	I*2	4-005404	IM	I*2	4-005402	IN	I*2	4-005400	J	I*2	4-005366	K	I*2	4-005336
LFLD	I*2	4-005364	LINE	I*2	9-000000	LSAV	I*2	4-005352	LSETS	I*2	4-005356	N	I*2	4-005376
NCH	I*2	4-005346	NLINS	I*2	4-005350	NMAX	I*2	4-005332	NN	I*2	4-005406	ØSETS	I*2	4-005360
PAGE	I*2	9-000002	PCCG1	R*4	6-003336	PCCG2	R*4	6-003342	PCC1	R*4	6-003326	PCC2	R*4	6-003332
PCLG1	R*4	6-003356	PCLG2	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	PI	I*2	4-005342
PSETS	I*2	4-005362	R	I*2	F-000004*	RJ	I*2	4-005370	SETSR	I*2	4-005372	TC	I*2	6-001554
T1	R*4	4-005326	XL	I*2	4-005354									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIØNS
A	R*4	6-001520	000030	12	(3,2)
BACAIN	R*4	6-000220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACØR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPØP	I*2	6-000000	000030	12	(12)
BARANS	R*4	6-001350	000060	24	(12)
BAUNCØ	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVU	I*2	6-003566	000200	64	(8,8)
BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUCØR	I*2	6-000170	000030	12	(12)

BULABL	L*1	6-001570	000010	4	(8)
BUPØP	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNCØ	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001300	000060	24	(12)
CCØNS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	000642	209	(418)
CLDØT	L*1	8-000000	000642	209	(418)
CLUD	R*4	6-004002	000010	4	(2)
IBUF	L*1	F-000002*	000001	0	(1)
LBLED	L*1	6-002773	000321	104	(209)
PF	L*1	4-005121	000204	66	(132)
PFN	L*1	4-000000	000204	66	(132)
PH	L*1	4-001020	004100	1056	(2112)
PN	L*1	4-000410	000204	66	(132)
PP	L*1	4-000204	000204	66	(132)
PTL	L*1	4-000614	000204	66	(132)
RCØNS	R*4	6-004112	001000	256	(2,8,8)
TCØNS	R*4	6-004012	000100	32	(2,8)
TYPE	L*1	6-002452	000321	104	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-001004	2	1-001426	3	1-002104	4	1-000724	5	**
6	1-002466	7	1-002240	8	1-000500	9	1-001464	10	1-001500
11	1-001632	12	1-000072	13	1-000100	14	1-001502	15	1-002666
16	1-000542	20	1-000630	22	1-001072	30	**	40	**
90'	3-000052	95'	3-000046	97'	3-000016	98'	3-000004	99'	3-000000
100'	3-000010	101'	3-000060	102'	3-000066	103'	3-000106	104'	3-000126

FUNCTIØNS AND SUBRØUTINES REFERENCED

BNT	CØRED	FNAME	KNT	MØTTL	MEAN	PØP	SNAME	STØMP

TØTAL SPACE ALLØCATED = 016640 3792

STDATA,LP:=STDATA

```
0001            SUBROUTINE DOTRAY(IBUF,A,FILE,PRTOUT)
0002           IMPLICIT INTEGER (A-Z)
0003            BYTE IBUF(1)
0004            INCLUDE 'COMMON.DAT'
0005  *        BYTE CLASFY(418),CLDAT(418)
0006  *         LOGICAL*1 TYPE(26),LABEL(5,9)
0007  *         LOGICAL*1 BALARL(26),BACLAS(26),BUCLAR(26),BULACS(26),GRAINS(26)
0008  *        REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
0009  *         REAL CLAD,CLUD,TCONS,RCONS,CCONS
0010  *         REAL RATCON,BUTCON,BARCON,BURCON
0011  *        DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,6,8)
0012  *         DIMENSION RATCON(2,6),BUTCON(2,6),BARCON(2,6,8),BURCON(2,6,8)
0013  *         DIMENSION PCVA(26,26),PCVL(26,26)
0014  *         DIMENSION BARBP(30),BUTBP(30)
0015  *         COMMON /CONAS/BARBP,BUTBP,BALARL,BACLAS,BUUNCO,BACOR,BUCOR,BACAIN,BUCAIN,
      *     *                   BAVAR,BUVAR,BARANS,BUTRNS,A,PRTOUT,TC,BUCATS,BUCLAR,
      *     *                   BULACS,BALACS,CLASFY,TYPE,LABEL,BACATS,BACLAR,AUNCO,
      *     *                   PCL1,PCL2,PCLG1,PCCG2,PCC1,PCC2,PCLG1,PCLG2,APOP,
      *     *                   PCVA,PCVL,RATBT,BUTBT,CLAD,CLUD,TCONS,RCONS,CCONS,
      *     *                   RATCON,BUTCON,BARCON,BURCON,BUTARG,GRAINS
0016  *         COMMON /DUMMY/CLADUM,CLUDUM
0017  *        COMMON/CLADAT/CLDAT
0018  *         REAL A(3,2)
0019  *         DIMENSION BATBP(30),BUTBP(30),BACOR(30),BUCOR(30)
0020  *         REAL BACAIN(26,26),BUCAIN(26,26),BAVAR(30),BUVAR(30)
0021  *         REAL BARANS(30),BUTRNS(30)
0022            COMMON/PRINT/LINE,PAGE
0023            REAL S(6),VC(2)
0024            INTEGER PAGE,FILE,R,REC,RC,SEISM,PRTOUT
0025            INTEGER I(2),J(3),K(3),PASS
0026            BYTE BLK,BLE,DOT,SLASH,TWO,LABEL(5)
0027            BYTE LC,CT(4)
0028            DATA BLE/' '/,SLASH/'/'/,TWO/'2'/
0029           DATA CT/' ','L','M','S'/
0030            DATA TH1/'T'/,TH2/'H'/
0031            DATA BLK/' '/
0032            LFLG=40
0033            K=1
0034            REC=1
0035            WRITE(5,100)
0036            WRITE(2,104) (I,I=1,4)
0037            WRITE(5,101)
0038            WRITE(5,102)
0039            WRITE(5,103)
0040            WRITE(5,105)
0041            LCNT=4
0042            SETS2=1.5
0043            CALL TIME
0044            READ(2,200,IOSTAT=IOS(5))SLC1
0045            READ(2,200,IOSTAT=IOS(7))SLC2
0046            READ(2,200,IOSTAT=IOS(11))SLC3
0047            READ(2,200,IOSTAT=IOS(14))SLC4
0048            IF(IOS(6).EQ.0)SLC1=SLC1-SLC2
0049            IF(IOS(8).EQ.0)SLC2=SLC2-SLC3
0050            IF(IOS(12).EQ.0)SLC3=SLC3-SLC4
0051            IF(IOS(15).EQ.0)SLC4=SLC4-SLC4
```

```
0052                  RJ=11
0053                  N=1
0054            1     CONTINUE
0055                  TYPE(K)=IBLF(RJ+12)
0056                  LABLE(K)=IBUF(RJ+11)
0057                  IF(IBUF(RJ+13) .EQ. '1') GO TO 8
0058                  CLASFY(2*K)=IBUF(RJ+9)
0059                  CLASFY(2*K-1)=IBUF(RJ+9)
0060                  GO TO 9
0061            8     CLASFY(2*K)=THR
0062                  CLASFY(2*K-1)=THR
0063          9       IF(LOOP.LT.ALL) GO TO 10
0064                  WRITE(3,103)
0065                  WRITE(3,104) (I,I=1,4)
0066                  WRITE(3,101)
0067                  WRITE(3,102)
0068                  WRITE(3,105)
0069                  WRITE(3,106)
0070                  LOOP=0
0071          10      DECODE(1,200,TYPE(K)) IU
0072                  IDET=1
0073                  DECODE(3,300,IBUF(RJ+29))GRN1
0074                  DECODE(3,300,IBUF(RJ+33))GRN2
0075                  DECODE(3,300,IBUF(RJ+37))GRN3
0076                  DECODE(3,300,IBUF(RJ+41))GRN4
0077                  IF(IBUF(RJ+29).EQ.'-')GRN1=0-GRN1
0078                  IF(IBUF(RJ+33).EQ.'-')GRN2=0-GRN2
0079                  IF(IBUF(RJ+36).EQ.'-')GRN3=0-GRN3
0080                  IF(IBUF(RJ+40).EQ.'-')GRN4=0-GRN4
0081                  GRN1=GRN1-SLR1
0082                  GRN2=GRN2-SLR2
0083                  GRN3=GRN3-SLR3
0084                  GRN4=GRN4-SLR4
0085                  WRITE(3,104) (IBUF(RJ+L-1),L=1,6),PT(10),LABLE(K),CLDET(2*K-1),
                    1CLDET(2*K),CLASFY(2*K-1),CLASFY(2*K),(IBUF(RJ+15+M),M=1,3),
                    2 GRN1,(IBUF(RJ+19+I),I=1,3),
                    3 GRN2,(IBUF(RJ+23+I),I=1,3),
                    4 GRN3,(IBUF(RJ+27+I),I=1,3), GRN4
0086                  LOOP=LOOP+1
0087                  K=K+1
0088                  IF(K.EQ.SETSP) GO TO 13
0089                  N=N+1
0090                  RJ=RJ+LFIL
0091                  GO TO 8
0092          13      IF(REC.EQ.14) GO TO 3
0093                  CALL GBRED(IBUF,8,FILE)
0094                  REC=REC+1
0095                  IF(REC.EQ.14) SETSP=14
0096                  GO TO 2
0097          3       PASS=2
0098                  CALL CALL
0099                  CALL DIAPRT(IBUF,TAB,PASS)
0100                  IF(PRTOUT.EQ.1) PRTOUT=
0101                  IF(CLAFOR.EQ.1) GO TO 20
0102                  IF(IFIX(1.1*SF.).LT.(IBUF(1)+1.1*F.)) PRTOUT=2
0103          20      IF(CLAFOR.EQ.1) GO TO
```

DRTRAY.FTN /TT:1/CS:5/E9

```
0104                 IF((IP.EQ.1).OR.(IP.EQ.2)) PRTOUT=2
0105                 RETURN
0106     100         FORMAT(1H1,53X,'DOT REPORT')
0107     101         FORMAT(' ',54X,4(' BRIGHT  GREEN'))
0108     102         FORMAT(' ',8X,'I  LINE  PIXEL  TYPE  LABEL  CLUSTER')
0109     103         FORMAT(1H+,49X,'CLASS',4('  -ESS    NO. '))
0110     104         FORMAT(' ',8X,3(I3,3A1),3X,2(I4,A1),1X,2(5X,A1,A1),
                    1 1X,4(I4,3A1,4X,I4))
0111     105         FORMAT(' ',1X)
0112     106         FORMAT('0',56X,4('ACQUISITION',I2,2X))
0113     200         FORMAT(I2)
0114     300         FORMAT(I3)
0115     400         FORMAT(I4)
0116                 END
```

PROGRAM SECTIONS

NUMBER	NAME	TYPE	ATTRIBUTES
1	SCODE1	002674	702	RW,I,CON,LCL
3	[illegible]	[illegible]	100	RW,D,LCL,LCL
4	[illegible]	[illegible]	77	RW,D,LCL,LCL
5	[illegible]	0000 0	1	RW,D,LCL,LCL
6	DRIVE	006542	5573	RW,D,OVR,GBL
7	[illegible]	000004	2	RW,D,OVR,GBL
8	CLOCK	000002	2*8	RW,D,OVR,GBL
9	PCNT	000004	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
DISPAY	I*2	0-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
APPR	I*2	0-016432	AREA	I*2	0-016436	BACATS	I*2	0-016032	BATAT	I*2	0-023352	BLK	I*1	0-000136
BUCATS	I*2	0-016076	BUTAT	I*2	0-023356	CLEARDW	I*2	7-000000	CLUMDUM	I*2	7-000002	DR	I*2	0-014200
DU	I*2	0-014202	ERLP	I*2	0-000060	GRAY	I*2	4-000206	GRID	I*2	4-000210	GRAY	I*2	4-000212
GRAY	I*2	4-000214	I	I*2	0-000000	IP	I*2	4-000224	K	I*2	4-000144	L	I*2	4-000216
LB	I*2	4-000222	LBUT	I*2	4-000176	LFLAT	I*2	4-000152	LIGHT	I*2	0-000000	M	I*2	4-000220
N	I*2	0-000002	TAST	I*2	0-000000	P	I*2	4-000232	PAGE	I*2	4-000147	PAGE0	I*1	4-000140
PAGE	I*2	2-000000	PASS	I*2	4-000134	PCC01	R*4	4-016100	PCC02	R*4	4-016104	PCC1	R*4	6-014070
PCC2	R*4	4-016120	PCL01	R*4	0-014200	PCL02	R*4	4-016124	PCL1	R*4	4-016110	PCL2	R*4	6-016114
PRINT	I*2	0-000000	Q	I*2	4-000224	R	I*2	0-000004	REC	I*2	4-000040	RJ	I*2	2-000042
SETSR	I*2	4-000000	SLASH	L*1	4-000141	SLG1	I*2	4-000172	SLG2	I*2	4-000174	SLG3	I*2	4-000176
SLG4	I*2	4-000200	TAPE	I*2	4-000150	TC	I*2	6-014214	TH1	I*2	4-000156	TH2	I*2	2-000160
TP	I*1	4-000000	U	I*2	4-000226									

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
A	R*4	0-014100	12	(3,2)
BACATS	R*4	0-000054	1352	(26,26)
BACLAY	L*1	4-016000	13	(26)
BATAT	I*2	0-000000	26	(26)
BAYAR	L*1	4-016000	13	(26)
BAY	I*2	4-000000	4	(2,2)
BARAY	R*4	4-016000	6	(3,2)
BARCOL	R*4	0-000000	676	(2,8,8)
BATCLA	L*1	0-000000	32	(2,16)
BAYCOL	I*2	0-000000	32	(32)
BAYAR	R*4	0-000000	64	(32)
BCC1	I*2	4-016000	676	(26,26)
BCV1	I*2	4-016000	676	(26,26)

BUCKET	I*2	A=[illegible]	[illegible]	16	(2,2)
BUCLAB	I*4	A=[illegible]	[illegible]	16	(2,2)
BUCNR	I*2	A=[illegible]	[illegible]	32	(32)
BULABL	I*4	A=[illegible]	[illegible]	32	(32)
BUMTP	I*2	A=[illegible]	[illegible]	32	(32)
BUSAVE	I*4	A=[illegible]	[illegible]	8	(32)
BUTAPE	I*2	A=[illegible]	[illegible]	256	(2,128)
BUTCB	I*4	A=[illegible]	[illegible]	16	(2,8)
BUVOCT	I*4	A=[illegible]	[illegible]	32	(32)
BUVAR	I*4	A=[illegible]	[illegible]	8	(8)
CCDUS	I*4	A=[illegible]	[illegible]	4	(2)
CLAP	I*4	A=[illegible]	[illegible]	4	(2)
CLASSFY	I*2	A=[illegible]	[illegible]	256	(128)
CLCAT	I*4	A=[illegible]	[illegible]	256	(128)
CLUD	I*4	A=[illegible]	[illegible]	4	(2)
CPL	I*2	A=[illegible]	[illegible]	10	(10)
DT	I*4	A=[illegible]	[illegible]	2	(2)
GRAINS	I*4	A=[illegible]	[illegible]	16	(2)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
LABEL	I*4	A=[illegible]	[illegible]	2	(2)
LALFD	I*4	A=[illegible]	[illegible]	128	(2,64)
LA	I*2	A=[illegible]	[illegible]	3	(3)
LT	I*2	A=[illegible]	[illegible]	5	(5)
RCP S	I*4	A=[illegible]	[illegible]	256	(2,128)
S	I*2	A=[illegible]	[illegible]	12	(12)
TC2S	I*4	A=[illegible]	[illegible]	32	(2,16)
TYPE	I*4	A=[illegible]	[illegible]	104	(52)
VC	I*4	A=[illegible]	[illegible]	4	(2)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1=[illegible]	2	1=[illegible]	3	1=[illegible]	8	1=[illegible]	9	1=[illegible]
10	1=[illegible]	11	1=[illegible]	20	1=[illegible]	1001	3=[illegible]	1011	3=[illegible]
1021	3=[illegible]	1031	3=[illegible]	1041	3=[illegible]	1051	3=[illegible]	1061	3=[illegible]
2001	3=[illegible]	3001	3=[illegible]	4001	**				

EQUIVALENCES AND SUBSTRING REFERENCED

DISPLACEMENT [illegible]

TOTAL SPACE ALLOCATED = [illegible]

ATTRIBUTES

```
@FORTRAN IV-PLUS V02-51                  08:04:13    09-MAR-78          PAGE 1
SEPRPT,FTN       /TR:3L:80:S/WR
0001             SUBROUTINE SEPRPT(TCHAN,T2TSEP,SDATA)
0002             IMPLICIT INTEGER(A-Z)
0003             LOGICAL*1 SDATA(1),T2TSEP(1)
0004             WRITE(2,200)
0005             IF(TCHAN.NE.0) GO TO 20
0006             WRITE(2,199)
0007     199     FORMAT('0','DATA FILE EMPTY')
0008             RETURN
0009     20      WRITE(2,201) (I,I=1,16)
0010             WRITE(2,202)
0011             WRITE(2,203) (T2TSEP(J),J=1,8)
0012             DPTR=0
0013             DO 50 I=1,14
0014             DO 47 KK=1,8
0015             IF(SDATA(KK+DPTR),NE,'0') GO TO 48
0016     47      CONTINUE
0017             GO TO 49
0018     48      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14),I
0019     1       WRITE(2,101)
0020             GO TO 46
0021     2       WRITE(2,102)
0022             GO TO 46
0023     3       WRITE(2,103)
0024             GO TO 46
0025     4       WRITE(2,104)
0026             GO TO 46
0027     5       WRITE(2,105)
0028             GO TO 46
0029     6       WRITE(2,106)
0030             GO TO 46
0031     7       WRITE(2,107)
0032             GO TO 46
0033     8       WRITE(2,108)
0034             GO TO 46
0035     9       WRITE(2,109)
0036             GO TO 46
0037     10      WRITE(2,110)
0038             GO TO 46
0039     11      WRITE(2,111)
0040             GO TO 46
0041     12      WRITE(2,112)
0042             GO TO 46
0043     13      WRITE(2,113)
0044             GO TO 46
0045     14      WRITE(2,114)
0046     46      WRITE(2,203) (SDATA(DPTR+KK),KK=1,8)
0047     49      DPTR=DPTR+8
0048     50      CONTINUE
0049             RETURN
0050     200     FORMAT(1H1,50X,'SEPARABILITY REPORT')
0051     201     FORMAT('0',19X,'CHANNELS',7X,16I4,6X,'SEPARABILITY')
0052     202     FORMAT('0',19X,'AVAILABLE',6X,<TCHAN>(2X,'X'))
0053     203     FORMAT(1H+,88X,2A1,',',6A1)
0054     101     FORMAT('0',19X,'1ARG',11X,4(2X,'X'))
0055     102     FORMAT('0',46X,4(2X,'X'))
0056     103     FORMAT('0',59X,4(2X,'X'))
```

SEPRPT.FTN /TR1BLOCKS/WR

```
0057     104      FØRMAT('0',70X,4(2X,'X'))
0058     105      FØRMAT('0',19X,'3ACØ',11X,8(2X,'X'))
0059     106      FØRMAT('0',34X,4(2X,'X'),12X,4(2X,'X'))
0060     107      FØRMAT('0',34X,4(2X,'X'),24X,4(2X,'X'))
0061     108      FØRMAT('0',46X,8(2X,'X'))
0062     109      FØRMAT('0',46X,4(2X,'X'),12X,4(2X,'X'))
0063     110      FØRMAT('0',58X,8(2X,'X'))
0064     111      FØRMAT('0',19X,'3ACØ',11X,12(2X,'X'))
0065     112      FØRMAT('0',36X,8(2X,'X'),12X,4(2X,'X'))
0066     113      FØRMAT('0',36X,4(2X,'X'),12X,8(2X,'X'))
0067     114      FØRMAT('0',46X,12(2X,'X'))
0068              END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001322	361	RW,I,CON,LCL
2	$DATA	000036	15	RW,D,CON,LCL
3	$IDATA	000640	208	RW,D,CON,LCL
4	$VARS	000010	4	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
SEPRPT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
DPTR	I*2	4-000004	I	I*2	4-000000	J	I*2	4-000002	KK	I*2	4-000006	TOTAL	I*2	3-000636

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
SDATA	L*1	F-000006*	000001	0	(1)
TOTSEP	L*1	F-000004*	000001	0	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000444	2	1-000476	3	1-000537	4	1-000569	5	1-000419
6	1-000642	7	1-000672	8	1-000729		1-000753	10	1-001089
11	1-001032	12	1-001062	13	1-001119	14	1-001143	29	1-000134
46	1-001170	47	**	48	1-000429	49	1-001254	50	**
101'	3-000174	102'	3-000222	103'	3-000248	104'	3-000256	105'	3-000274
106'	3-000322	107'	3-000352	108'	3-000402	109'	3-000420	110'	3-000451
111'	3-000466	112'	3-000514	113'	3-000544	114'	3-000571	199'	3-000608
200'	3-000024	201'	3-000056	202'	3-000122	203'	3-000154		

TOTAL SPACE ALLOCATED = 002230 588

NO FPP INSTRUCTIONS GENERATED

SEPRPT,LP1=SEPRPT

```
 #FORTRAN IV-PLUS V02-51                    08:04:25    09-MAR-78          PAGE 1
CALC,FTN        /TR:BL:CAS/WF
0001            SUBROUTINE CALC
0002            IMPLICIT INTEGER(A-Z)
0003            INCLUDE 'COMMON1.DAT'
0004 *          BYTE CLASFY(418),CLDAT(418)
0005 *          LOGICAL*1 TYPE(209),LBLED(209)
0006 *          LOGICAL*1 BALABL(8),BACLAB(8),BLCLAB(8),BULABL(8)
0007 *          REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
0008 *          REAL CLAD,CLUD,TCANS,RCANS,CCANS
0009 *          DIMENSION CLAD(2),CLUD(2),TCANS(2,8),CCANS(2),RCANS(2,8,8)
0010 *          DIMENSION PCVA(8,8),PCVU(8,8)
0011 *          DIMENSION BAPOP(12),BUPOP(12)
0012 *          COMMON/CBIG/BAPOP,BUPOP,BALNCO,BUUNCO,BACOR,BUCOR,BACAIN,BUCAIN,
     *     *                BAVAR,BUVAR,BARANS,BURANS,A,DD,DU,TC,BUCATS,BUCLAB,
     *     *                BULABL,BALABL,CLASFY,TYPE,LBLED,BACATS,BACLAB,
     *     *                PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
     *     *                PCVA,PCVU,BATOT,BLTOT,CLAD,CLUD,TCANS,RCANS,CCANS
0013 *          COMMON/DUMMY/CLADUM,CLUDUM
0014 *          COMMON/CLCOM/CLDAT
0015 *          REAL A(3,2)
0016 *          DIMENSION BAUNCO(12),BUUNCO(12),BACOR(12),BUCOR(12)
0017 *          REAL BACAIN(8,8),BUCAIN(8,8),BAVAR(12),BUVAR(12),BARANS(12),BURANS(12)
0018            DIMENSION LCAT(11),BCVT(8)
0019            REAL BSUM,VAR,ALN,ALG,BASE1,BASE2,N3F,BASEF
0020            REAL ALNTF,ALNBF,ALGTF,ALGBF,BAPF,GRSF,Z4F
0021            REAL N2CLF,LCATF(12),SLN,SLG,BARY(8),BAPOPF(12)
0022            BYTE LB,CL,C(12),CL2
0023            NTYP2=0
0024            NTYP1=0
0025            NAII=0
0026            NBII=0
0027            NGIJ=0
0028            BASE=32932
0029            GWS=0
0030            BAPOP(12)=0
0031            BAVAR(12)=0
0032            BACOR(12)=0
0033            BAUNCO(12)=0
0034                 BARANS(12)=0
0035            N=0
0036            S=0
0037            W=0
0038            CHECK=0
0039            PCC1=0
0040            PCC2=0
0041            N2CL=0
0042            PCCG1=0
0043            PCCG2=0
0044            IF(CLADUM.EQ.1) GO TO 475
0045            DO 15 I=1,BACATS
0046            IF(BACLAB(I).NE.'W') GO TO 10
0047            CHECK=CHECK+1
0048            W=I
0049            GO TO 15
0050      10    IF(BACLAB(I).NE.'S') GO TO 12
0051            CHECK=CHECK+1
0052            S=I
```

```
0053                GØ TØ 15
0054     12         IF(BACLAR(I),NE,'N') GØ TØ 15
0055                N=1
0056     15         CØNTINUE
0057                DØ 40 I=1,11
0058                LCAT(I)=0
0059     40         CØNTINUE
0060                DØ 401 J=1,8
0061                DØ 401 I=1,8
0062                RCVA(J,I)=0
0063     401          CØNTINUE
0064                DØ 41 I=1,12
0065                PARAMS(I)=0
0066     41         CØNTINUE
0067                DØ 39 I=1,202
0068                LB=LABELED(I)
0069                CL=CLASFY(2*I-1)
0070                CL2=CLASFY(2*I)
0071                IF(LB .EQ. ' ') GØ TØ 39
0072                IF (CL .EQ.'0' .ØR.CL.EQ.' ') GØ TØ 39
0073                IF (TYPE(I).EQ.'2'.ØR.TYPE(I).EQ.'0') GØ TØ 35
                C
                C CØMPUTE TYPE 1 DØT DATA
                C
0074                NTYP1=NTYP1+1
0075                IF (LB.NE.CL .AND. CL.NE.' ') GØ TØ 32
0076                NAII=NAII +1
0077                GØ TØ 35
0078     32         IF(CHECK.LT.2)GØ TØ 35
0079     33         IF (LB.EQ.BACLAB(W).AND.CL.EQ.BACLAB(S)) NGIJ=NGIJ+1
0080                IF(LB.EQ.BACLAB(S).AND.CL.EQ.BACLAB(W)) NGIJ=NGIJ+1
                C
                C CØMPUTE TYPE 2 DØT
                C
0081     35         IF (TYPE(I).NE.'2')GØ TØ 39
0082                IF(CL2.NE.' ') GØ TØ 39
0083                NTYP2=NTYP2+1
0084                IF(LB.NE.' '.AND.CL.NE.' ') NØCL=NØCL+1
                C
                C CØNSTRUCT BIAS CØRRECTIØN VECTØRS(LB)
                C
0085                I1=0
0086                I2=0
0087                DØ 43 K=1,NACATS
0088                IF(LB.EQ.BACLAB(K))I1=K
0089                IF(CL.EQ.BACLAB(K))I2=K
0090     43         CØNTINUE
0091                IF(I1.EQ.0.ØR.I2.EQ.0)GØ TØ 39
0092                RCVA(I1,I2)=RCVA(I1,I2)+1
0093     39         CØNTINUE
                C
                C CØMPUTE GRAIN PØPULATIØN, CLASSIFIED %, CØRRECTED %
                C
0094                IF(CHECK.LT.2)GØ TØ 25
0095                BAPØP(12)=BAPØP(W)+BAPØP(S)
0096                BAUNCØ(12)=BAUNCØ(W)+BAUNCØ(S)
```

```
                C
                C COMPUTE GRAIN CORRECTED % AND VARIANCE %
                C
0097        47      ALGT=BCVA(W,W)+BCVA(W,S)+BCVA(S,W)+BCVA(S,S)
0098                ALGB=0
0099                DØ 155 I=1,BACATS
0100                ALGB=ALGB+BCVA(I,W)+BCVA(I,S)
0101        155     CONTINUE
0102                ALGTF=FLØAT(ALGT)
0103                ALGBF=FLØAT(ALGB)
0104                ALG=ALGTF/ALGBF
0105                ALNB=0
0106                ALNT=0
0107                DØ 156 I=1,BACATS
0108                IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 156
0109                DØ 156 J=1,BACATS
0110                IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 156
0111                ALNT=ALNT+BCVA(I,J)
0112        156     CØNTINUE
0113                DØ 158 I=1,BACATS
0114                DØ 158 J=1,BACATS
0115                    IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 158
0116                ALNB=ALNB+BCVA(I,J)
0117        158     CØNTINUE
0118                BASEF=FLØAT(BASE-BAPØP(11))
0119                BASE1=0.
0120                IF(BASEF.LE.0) BASEF=1
0121                BAPF=FLØAT(BAPØP(12))
0122                BASE1=BAPF/BASEF
0123                ALNTF=FLØAT(ALNT)
0124                ALNBF=FLØAT(ALNB)
0125                IF(ALNBF.LE.0) ALNBF=1
0126                ALN=ALNTF/ALNBF
0127                N3F=0.
0128                DØ 159 I=1,BACATS
0129                IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 159
0130                N3F=N3F+BAPØP(I)
0131        159     CØNTINUE
0132                BACØP(12)=(ALG*BASE1+(1.-ALN)*N3F/BASEF)*100.
0133                BASE2=(N3F/BASEF*100.)**2
0134                Z4=0
0135                DØ 21 J=1,BACATS
0136                Z4=Z4+BCVA(J,W)+BCVA(J,S)
0137        21      CØNTINUE
0138                VAR=0.
0139                Z4F=FLØAT(Z4-1)
0140                IF (Z4.LT.1)GØ TØ 24
0141                VAR=((BASE1*100.)**2*(ALG*(1.-ALG)))/Z4F
0142        24      IF(ALNBF .GT. 1.) GØ TØ 22
0143                BAVAR(12)=VAR
0144                GØ TØ 25
0145        22      BAVAR(12)=VAR+BASE2*ALN*(1.-ALN)/(ALNBF-1.)
                C
                C COMPUTE THE RANDØM SAMPLE FØR EACH CATEGØRY EXCEPT FØR GRAIN
                C
0146        25          BATØT=0
```

```
CALC,FTN        /TR:PLØCKS/WR
0147                PSUM=0.
0148                DØ 160 I=1,BACATS
0149                PSUM=PSUM+FLØAT(-APØP(I))
0150    160             CØNTINUE
0151                PSUM=PSUM/(22932.+APØP(11))
0152                NØCLF=FLØAT(NØCL)
0153                DØ 165 I=1,BACATS
0154                BCVT(I)=0
0155                DØ 168 J=1,BACATS
0156                BCVT(I)=BCVT(I)+BCVA(I,J)
0157    168         CØNTINUE
0158                BARANS(I)=FLØAT(BCVT(I))/NØCLF*PSUM*100.
0159                BATØT=BATØT+BCVT(I)
0160                IF(I.NE.W.AND.I.NE.S)GØ TØ 165
0161                GWS=GWS+BCVT(I)
0162    165         CØNTINUE
                    C
                    C COMPUTE THE RANDØM SAMPLE FØR GRAIN
                    C
0163                IF(CHECK.LT.2)GØTØ 100
0164                GWSF=FLØAT(GWS)
0165                BARANS(12)=GWSF/NØCLF*PSUM*100.
                    C
                    C COMPUTE PCC1, PCC2, PCCG1, PCCG2
0166    100         BATAT=0
0167                X=0
0168                IF (NTYP1 .EQ. 0) GØ TØ 45
0169                PCC1=FLØAT(NAIT)*100./FLØAT(NTYP1)
0170    45          IF(NTYP2 .EQ. 0) GØ TØ 469
0171                X=BCVA(1,1)+BCVA(2,2)+BCVA(3,3)+BCVA(4,4)+BCVA(5,5)
0172                X=X+BCVA(6,6)+BCVA(7,7)+BCVA(8,8)
0173                PCC2=FLØAT(X)*100./FLØAT(NTYP2)
0174    469         IF(CHECK.LT.2)GØ TØ 470
0175                BATAT=BCVA(W,S)+BCVA(S,W)
0176                DØ 175 I=1,BACATS
0177                BATAT=BATAT+BCVA(I,I)
0178    175         CØNTINUE
0179                    IF (NTYP1.EQ.0)GØ TØ 468
0180                PCCG1=(FLØAT(NAIT)+FLØAT(NGIJ))*100./FLØAT(NTYP1)
0181    468         IF(NTYP2.EQ.0)GØ TØ 470
0182                PCCG2=FLØAT(BATAT)*100./FLØAT(NTYP2)
0183    470         CØNTINUE
                    C CLADW CALCULATIØNS FIRST
0184                DØ 280 MM=1,2
0185                IF(MM.EQ.1) KK=W
0186                IF(MM.EQ.2) KK=S
0187                SLW=0.
0188                CLW=0.
0189                IF(KK.EQ.0) GØ TØ 270
0190                WARY(7)=0.
0191                WARY(1)=RACØR(KK)
0192                WARY(2)=BAUNCØ(KK)
0193                WARY(3)=BARANS(KK)
0194                WARY(4)=RAVAR(KK)
0195                WARY(5)=RACAIN(KK,KK)
0196                WARY(6)=PCC1
```

```
CALC.FTN        /TR1BLØCKS/WR
0197            IF(N.NE.0) WARY(7)=BACAIN(N,N)
0198             ARY(8)=PC**2
0199            DØ 250 I=1,8
0200            SLW=SLW+WARY(I)*TCØNS(MM,I)
0201            DØ 240 J=1,8
0202            CLW=CLW+WARY(I)*WARY(J)*RCØNS(MM,I,J)
0203    240     CØNTINUE
0204    250     CØNTINUE
0205            SLW=SLW+CCØNS(MM)
0206    270     CLAD(MM)=CLW+SLW
0207    280     CØNTINUE
0208            NTYP2=0
0209            NAII=0
0210            NBII=0
0211            NGIJ=0
0212            BASE=22932
0213            GWS=0
0214            BUP2P(12)=0
0215           BUVAR(12)=0
0216            BUC2R(12)=0
0217            BUUNCØ(12)=0
0218                 BURANS(12)=0
0219            N=0
0220            S=0
0221            CHECK=0
0222            PCL1=0
0223            NØCL=0
0224            NTYP1=0
0225            PCL2=0
0226            PCLG1=0.
0227            PCLG2=0.
0228            IF(CLUDUM.EQ.1) RETURN
0229            DØ 915 I=1,BUCATS
0230            IF(BUCLAR(I).NE.'N') GØ TØ 910
0231            CHECK=CHECK+1
0232            N=I
0233            GØ TØ 915
0234    910     IF(BUCLAR(I).NE.'S') GØ TØ 915
0235            CHECK=CHECK+1
0236            S=I
0237    915     CØNTINUE
0238            DØ 940 I=1,11
0239            LCAT(I)=0
0240    940     CØNTINUE
0241            DØ 9401 J=1,8
0242            DØ 9401 I=1,8
0243            BCVU(J,I)=0
0244    9401      CØNTINUE
0245            DØ 941 I=1,12
0246            BURANS(I)=0
0247    941     CØNTINUE
0248            DØ 939 I=1,209
0249            LB=LABLED(I)
0250            CL=CLDØT(2*I-1)
0251            CL2=CLDØT(2*I)
0252            IF(LB.EQ.' ') GØ TØ 939
```

```
CALC.FTN        /TR:BLØCKS/WD
0253            IF (CL .EQ.'0' .OR.CL.EQ.' ') GØ TØ 939
0254            IF (TYPE(I).EQ.'2'.ØR.TYPE(I).EQ.'D') GØ TØ 935
                C
                C CØMPUTE TYPE 1 DØT DATA
                C
0255            NTYP1=NTYP1+1
0256            IF (LB.NE.CL .AND. CL.NE.' ') GØ TØ 932
0257            NAII=NAII +1
0258            GØ TØ 935
0259    932     IF(CHECK.LT.2)GØ TØ 935
0260    933     IF (LB.EQ.BUCLAB(W).AND.CL.EQ.BUCLAB(S)) NGIJ=NGIJ+1
0261            IF(LB.EQ.BUCLAB(S).AND.CL.EQ.BUCLAB(W)) NGIJ=NGIJ+1
                C
                C CØMPUTE TYPE 2 DØT
                C
0262    935     IF (TYPE(I).NE.'2')GØ TØ 939
0263            IF(CL2.NE.' ') GØ TØ 939
0264            NTYP2=NTYP2+1
0265            IF(LB.NE.' '.AND.CL.NE.' ') N2CL=NØCL+1
                C
                C CØNSTRUCT BIAS CØRRECTIØN VECTØRS(LB)
                C
0266            Z1=0
0267            Z2=0
0268            DØ 943 K=1,BUCATS
0269            IF(LB.EQ.BUCLAB(K))Z1=K
0270            IF(CL.EQ.BUCLAB(K))Z2=K
0271    943     CØNTINUE
0272            IF(Z1.EQ.0.ØR.Z2.EQ.0)GØ TØ 939
0273            BCVU(Z1,Z2)=BCVU(Z1,Z2)+1
0274    939     CØNTINUE
                C
                C CØMPUTE GRAIN PØPULATIØN, CLASSIFIED %, CØRRECTED %
                C
0275            IF(CHECK.LT.2)GØ TØ 925
0276            BUPØP(12)=BUPØP(W)+BUPØP(S)
0277            BUUNCØ(12)=BUUNCØ(W)+BUUNCØ(S)
                C
                C CØMPUTE GRAIN CØRRECTED % AND VARIANCE %
                C
0278    947       ALGT=BCVU(W,W)+BCVU(W,S)+BCVU(S,W)+BCVU(S,S)
0279            ALGR=0
0280            DØ 9155 I=1,BUCATS
0281            ALGR=ALGR+BCVU(I,W)+BCVU(I,S)
0282    9155    CØNTINUE
0283            ALGTF=FLØAT(ALGT)
0284            ALGRF=FLØAT(ALGR)
0285            ALG=ALGTF/ALGRF
0286            ALNR=0
0287            ALNT=0
0288            DØ 9156 I=1,BUCATS
0289            IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 9156
0290            DØ 9156 J=1,BUCATS
0291            IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 9156
0292            ALNT=ALNT+BCVU(I,J)
0293    9156    CØNTINUE
```

```
CALC,FTN          /TR:BLØCKS/WP
0294              DØ 9158 I=1,NUCATS
0295              DØ 9158 J=1,NUCATS
0296                 IF(J.EQ.N.ØR.J.EQ.S)GØ TØ 9158
0297              ALNR=ALNR+RCVU(I,J)
0298    9158      CØNTINUE
0299              BASEF=FLØAT(BASE-BUP0P(11))
0300              BASE1=0.
0301              IF(BASEF.LE.0) BASEF=1
0302              BAPF=FLØAT(BUP0P(12))
0303              BASE1=BAPF/BASEF
0304              ALNTF=FLØAT(ALNT)
0305              ALNBF=FLØAT(ALNB)
0306              IF(ALNBF.LE.0) ALNBF=1
0307              ALN=ALNTF/ALNBF
0308              N3F=0.
0309              DØ 9159 I=1,NUCATS
0310              IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 9159
0311                 N3F=N3F+FLØAT(BUP0P(I))
0312    9159      CØNTINUE
0313              BUCVR(12)=(ALG*(BASE1+(1.-ALN)*N3F/BASEF)*100.
0314              BASE2=(N3F/BASEF*100.)**2
0315              Z4=0
0316              DØ 924 J=1,NUCATS
0317              Z4=Z4+RCVU(J,W)+RCVU(J,S)
0318    924       CØNTINUE
0319              VAR=0.
0320              Z4F=FLØAT(Z4-1)
0321              IF (Z4.LT.1)GØ TØ 924
0322              VAR=((BASE1*100.)**2*(ALG*(1.-ALG)))/Z4F
0323              IF(ALNBF .GT. 1.) GØ TØ 922
0324              BUVAR(12)=VAR
0325              GØ TØ 925
0326    922       BUVAR(12)=VAR+BASE2*ALN*(1.-ALN)/(ALNBF-1.)
                  C
                  C CØMPUTE THE RANDØM SAMPLE FØR EACH CATEGØRY EXCEPT FØR GRAIN
                  C
0327    925               BUTØT=0
0328              PSUM=0.
0329              DØ 9160 I=1,NUCATS
0330              PSUM=PSUM+FLØAT(BUP0P(I))
0331    9160         CØNTINUE
0332              PSUM=PSUM/(22932.-BUP0P(11))
0333              NCLF=FLØAT(NCL)
0334              DØ 9165 I=1,NUCATS
0335              RCVT(I)=0
0336              DØ 9168 J=1,NUCATS
0337              RCVT(I)=RCVT(I)+RCVU(I,J)
0338    9168      CØNTINUE
0339              BURAS(I)=FLØAT(RCVT(I))/NCLF*PSUM*100.
0340              BUTØT=BUTØT+RCVT(I)
0341              IF(I.NE.W.AND.I.NE.S)GØ TØ 9165
0342              GWS=GWS+RCVT(I)
0343    9165      CØNTINUE
                  C
                  C CØMPUTE THE RANDØM SAMPLE FØR GRAIN
                  C
```

```
CALC,FTN        /TR;BLOCKS/WR
0344            IF(CHECK.LT.2)GO TO 9100
0345            GWSF=FLOAT(GWS)
0346            RUNOFF(12)=GWSF/AOCLF*PSUM*100.
                C
                C COMPUTE PCC1, PCC2, PCCG1, PCCG2
0347    9100    RATAT=0
0348            X=0
0349            IF (NTYP1 .EQ. 0) GO TO 945
0350            PCL1=FLOAT(NAII)*100./FLOAT(NTYP1)
0351    945     IF(NTYP2 .EQ. 0) GO TO 9469
0352            X=RCVU(1,1)+RCVU(2,2)+RCVU(3,3)+RCVU(4,4)+RCVU(5,5)
0353            X=X+RCVU(6,6)+RCVU(7,7)+RCVU(8,8)
0354            PCL2=FLOAT(X)*100./FLOAT(NTYP2)
0355    9469     IF(CHECK.LT.2)GO TO 9470
0356            RATAT=RCVU(W,S)+RCVU(S,W)
0357            DO 9175 I=1,RUCATS
0358            RATAT=RATAT+RCVU(I,I)
0359    9175    CONTINUE
0360                IF (NTYP1.EQ.0)GO TO 9468
0361            PCLG1=(FLOAT(NAII)+FLOAT(NGIJ))*100./FLOAT(NTYP1)
0362    9468     IF(NTYP2.EQ.0)GO TO 9470
0363            PCLG2=FLOAT(RATAT)*100./FLOAT(NTYP2)
0364    9470    CONTINUE
                C CLOUD CALCULATIONS NOW
0365            DO 9260 MM=1,2
0366            IF(MM.EQ.1) KK=W
0367            IF(MM.EQ.2) KK=S
0368            SLW=0.
0369            CLW=0.
0370            IF(KK.EQ.0) GO TO 9270
0371            WARY(7)=0.
0372            WARY(1)=RUCOV(KK)
0373            WARY(2)=RUINC2(KK)
0374            WARY(3)=RUPAVS(KK)
0375            WARY(4)=RUVAR(KK)
0376            WARY(5)=RUCAIN(KK,KK)
0377            WARY(6)=PCL1
0378            IF(N.NE.0) WARY(7)=BUCAIN(N,N)
0379            WARY(8)=PCL2
0380            DO 9250 I=1,8
0381            SLW=SLW+WARY(I)*TCONS(MM,I)
0382            DO 9240 J=1,8
0383            CLW=CLW+WARY(I)*WARY(J)*RCONS(MM,I,J)
0384    9240    CONTINUE
0385    9250    CONTINUE
0386            SLW=SLW+CCONS(MM)
0387    9270    CLUD(MM)=CLW+SLW
0388    9280    CONTINUE
0389            CONTINUE
0390            RETURN
0391            END
```

PROGRAM SECTION'S

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	011150	2376	RW,I,CON,LCL
2	$PDATA	000004	2	RW,D,CON,LCL
4	$VARS	000462	153	RW,D,CON,LCL
5	$TEMPS	000334	14	RW,D,CON,LCL
6	OTHER	000122	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLOCK	000642	209	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CALC		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALG	R*4	4-000060	ALGF	I*2	4-000442	ALGHF	R*4	4-000122	ALN	I*2	4-000444	ALNF	I*2	4-000116
ALNHF	R*4	4-000156	ALPH	I*2	4-000446	ALPHHF	R*4	4-000112	ALT	I*2	4-000450	ALTF	I*2	4-000176
BACATS	I*2	6-003314	BASE	R*4	4-000120	BASE	I*2	4-000452	BASEF	I*2	4-000454	BASE1	R*4	4-000346
BASE2	R*4	4-000172	BATAT	I*2	4-000456	BATFT	I*2	6-003766	BICATS	I*2	6-003770	BIT	I*2	6-003772
CHECK	I*2	4-000422	CL	I*1	4-000357	CLARGV	I*2	7-000000	CLIOPT	I*2	7-000002	CM	I*4	4-000232
CL2	L*1	4-000374	CP	I*2	4-001550	DJ	I*2	6-001552	DLE	I*2	4-000402	DLF	R*4	4-000132
I	I*2	4-000426	J	I*2	4-000430	K	I*2	4-000434	KK	I*2	4-000440	LF	I*1	4-000356
MM	I*2	4-000436	N	I*2	4-000438	NAII	I*2	6-000442	NPTT	I*2	4-000444	NT	I*2	4-000406
NCOL	I*2	4-000424	NDCLF	R*4	4-000142	NTYP1	I*2	4-000400	NTYP2	I*2	4-000402	NU	R*4	4-000076
PCC01	R*4	6-003336	PCC02	R*4	6-003342	PCC1	R*4	6-003324	PCC2	R*4	6-003332	PCL01	R*4	6-003356
PCL02	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	PSLP	R*4	4-000410	S	I*2	4-000416
SL	R*4	4-000226	TT	I*2	6-001554	VAR	R*4	4-000032	X	I*2	4-000454	Y	I*2	6-000454
Z1	I*2	4-000432	Z2	I*2	4-000434	Z4	I*2	4-000450	Z4F	I*2	4-000452			

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
BACATS	R*4	6-003220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACOR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPRP	I*2	6-000000	000030	12	(12)
BAPRPF	R*4	4-000276	000060	24	(12)
BARAYS	R*4	6-001350	000060	24	(12)
BAUNIT	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVT	I*2	4-000024	000020	8	(8)
BCVU	I*2	6-003566	000200	64	(8,8)

BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUC2R	I*2	6-000170	000030	12	(12)
BULABL	L*1	6-001570	000010	4	(8)
BUP2P	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNCO	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001310	000060	24	(12)
C	L*1	4-000350	000014	6	(12)
CCONS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	001642	209	(41B)
CLDAT	L*1	6-000000	001642	209	(41B)
CLUD	R*4	6-004002	000010	4	(2)
LBLED	L*1	6-002773	000321	104	(209)
LCAT	I*2	4-000000	000026	11	(11)
LCATF	R*4	4-000146	000060	24	(12)
RCONS	R*4	6-004112	001000	256	(2,8,8)
TCONS	R*4	6-004012	000100	32	(2,8)
TYPE	L*1	6-002452	000321	104	(209)
VARY	R*4	4-000236	000040	16	(8)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000236	12	1-000272	15	1-000320	21	**	22	1-002450
24	1-002612	25	1-002724	32	1-000702	33	**	35	1-006100
39	1-001210	40	**	41	**	47	**	48	1-003364
47	**	100	1-003312	155	**	156	1-001460	156	1-002034
159	1-002302	160	**	165	1-003224	168	**	175	**
240	**	250	**	270	1-004440	280	**	401	**
468	1-003570	469	1-003674	470	1-003732	910	1-004726	915	1-004752
921	**	922	1-007302	924	1-007244	925	1-007356	932	1-005134
933	**	935	1-005432	939	1-005642	940	**	941	**
943	**	945	1-010016	947	**	9101	1-007744	9105	**
9156	1-006320	9158	1-006466	9159	1-006734	9160	**	9165	1-007654
9168	**	9175	**	9240	**	9250	**	9370	1-010272
9280	**	9401	**	9468	1-010322	9469	1-010424	9470	1-010364

TOTAL SPACE ALLOCATED = 017662 4057

CALC,LP:=CALC

```
*FORTRAN IV-PLUS V02-51                    08:05:19    09-MAR-78          PAGE 1
BIAPRT.FTN      /TR:PL:DC/S/WR
0001            SUBROUTINE BIAPRT(IUNF,TYPE,PASS)
0002            IMPLICIT INTEGER (A-Z)
0003            LOGICAL* 1 TYPE(1)
0004            INCLUDE 'FGROUP.CMN/LIST'
0005 *          BYTE CLASFY(4,8),CLPRT(418)
0006 *          LOGICAL* 1 TYPE(2,9),LBLAB(209)
0007 *          LOGICAL* 1 BALABL(8),PACLAB(8),PLTLAB(8),PULABL(8)
0008 *          REAL PCC1,PCC2,PCCS1,PCCS2,PCL1,PCL2,PCLS1,PCLS2
0009 *          REAL CLAT,CLONT,TOTS,PCTS,CCTS
0010 *          INTEGER CLAT(2),CLONT(2),TOTS(2,8),CCTS(2),PCTS(2,8,8)
0011 *          DIMENSION CLASS(2,21,8,8)
0012 *          DIMENSION APCP(12),ASPCP(12)
0013 *          COMMON /CRIAS/NAPPR,NSPPR,BALICE,BUOYZ,EACTR,RUCAN,RICAT,PLCAT,
     *                  1AMAP,PLLAB,PSPARS,PLPARS,LANS,LCR,DI,PC,PLCTYP,NCLTR,
     *                  2CLASFY,BALABL,CLASFY,TYPE,LBLAB,ACATS,PLCLAB,
     *                  3PCC1,PCC2,PCCS1,PCCS2,PCL1,PCL2,PCLS1,PCLS2,
     *                  4CLAT,CLONT,CLPRT,CLSD,CLUP,ICR,PCTS,CCTS
0014 *          COMMON /DIMTY/CLAT,CLONT
0015 *          COMMON /CLDZN/CLASS
0016 *          REAL X(3,2)
0017 *          DIMENSION TAPCC(12),PUOYCP(12),PAPCP(12),PUOYP(12)
0018 *          REAL EACTR(5,5),RUCAN(5,5),EVAR(12),ELVAR(12),EARR S(12),PLCR S(12)
0019            COMMON /S-G/SEGS
0020            LOGICAL* SEGS(8),P(120),ELANK,PMATR(120)
0021            DIMENSION CL(10)
0022            DATA CL/10,20,30,40,5 ,60,70,80,90,100,110,120,130,140,
     *                  150,160,170,180,190/
0023            DATA BLANK/' '/,ONE/'1'/,SLASH/'/'/,TWO/'2'/
0024      100   FORMAT(1H1,45X,'BIAS CORRECTION CLASSIFICATION REPORT')
0025      101   FORMAT(1HO,2PX,12(5X,'CAT'))
0026      102   FORMAT(1H ,15X,5(7X,'A'))
0027      103   FORMAT(1HO,5X,'P FOR CAT'     ',12(3X,I5))
0028      104   FORMAT(1H ,5X,'CLASSIFIED    % ',12(2X,I6))
0029      105   FORMAT(1H ,5X,'CORRECTED     % ',8(9X,I6),2(4X,I6))
0030      106   FORMAT(1H ,5X,'VARIANCE      % ',8(2X,F6.2),2(4X,F6.2)
0031      107   FORMAT(1H ,5X,'MEANS OF SAMPLE  ',8(2X,F6.2),2(4X,F6.2)
0032      108   FORMAT(1HO,51X,'FLAG VALUE MATRIX')
0033      109   FORMAT(1H ,15X,8(5X,'CAT'))
0034      110   FORMAT(1H ,5X,'CAT. OF INT.'/,5(4X,I1,1X))
0035      111   FORMAT(1H ,5X,'CAT. OF INT.')
0036      112   FORMAT(1H ,11X,I1,5X,8(4X,F4.3))
0037      113   FORMAT(1HO,3PX,'BIAS CORRECTION VECTORS (LABLED:CLASSIFIED)')
0038      114   FORMAT(1HO,14X,5(4X,1A1,':',I1),I3)
0039      115   FORMAT('1',40X,'BIAS INTERFACE REPORT')
0040      116   FORMAT(1HO,3X,'LOAD ID  = ',24A1,7X,'TAPE NUMBER: ',I5,'01')
0041      117   FORMAT(1HO,3X,'RECORD ID = ',6A1,2X,6A1)
0042      118   FORMAT(1HO,5X,'ACQUISITION DATE')
0043      119   FORMAT(1H ,47X,'1',5X,'2',5X,'3',6X,'4')
0044      120   FORMAT(1H ,4X,'******  SEGMENT NUMBER =',4A1,13X,4(5:1,2X))
0045      121   FORMAT(1H ,'TRAINING SEGMENT NO. ',I2,' =',8A1)
0046      122   FORMAT(1HO,42X,'TYPE',1A1,' DAT LABEL/CLASSIFICATION')
0047      123   FORMAT(1HO,4 X,'TYPE',1A1,' DAT LABEL/CL STER ASSIGNMENT')
0048      124   FORMAT(1HO,1X,'BLANK')
0049      125   FORMAT(' ',1X,'ALL',2X,10(I3,3X))
0050      126   FORMAT(1H+,97X,'MEAN',4X,'VAR',6X,'MIN',4X,'MAX',4X,'ERROR')
0051      127   FORMAT(1H+,97X,'MEAN',4X,'VAR',6X,'MIN',4X,'MAX',4X,'ERROR')
```

```
FORTRAN IV-PLUS V02-51          08:05:19    09-MAR-79          PAGE 2
BIAPRT.FTN      /TR:SLOCKS/WD
0052      128   FORMAT(1H1,41X,'BIAS CORRECTION CLUSTER REPORT')
0053      129   FORMAT(1H1,13X)
0054      130   FORMAT(1H+,93X,'PCC1  = ',F6.1,' PCC2  = ',F6.1)
0055      131   FORMAT(1H+,93X,'PCCG1 = ',F6.1,' PCCG2 = ',F6.1)
0056      136   FORMAT(1H+,93X,' DW   = ',F6.1,'  DS   = ',F4.1)
0057      132   FORMAT(1H0,13X)
0058      133   FORMAT(1H+,70(,'CLUSTERS ')
0059      134   FORMAT(' ',20X,8(A1,'I',A1,I4,4X))
0060      135   FORMAT(1H0,22X,'TOTAL = ',I5)
0061            GO TO(1,2),PASS
0062        1   WRITE(6,115)
0063            DO 3 I=1,4
0064      3     SECNO(I)= BUF(I+4)
0065            WRITE(6,116) (IBUF(I),I=3,26),TSOV
0066            WRITE(6,117) (IBUF(I),I=709,714),(IBUF(J),J=715,720)
0067            WRITE(6,118)
0068            WRITE(6,119)
0069            WRITE(6,120) (SECNO(I),I=1,4),(IBUF(J),J=156,158),
           2(IBUF(K),K=159,163),(IBUF(L),L=164,168),(IBUF(M),M=169,173)
0070            RETURN
0071      2     IF(CLASOV.EQ.1) GO TO 2_2
0072            WRITE(6,129)
0073            CNT=ONE
0074            DO 97 JJ=1,2
0075            K=1
0076            NLINE=10
0077            WRITE(6,122) CNT
0078            WRITE(6,124)
0079            WRITE(6,125) (COL(I),I=1,15)
0080            DO 98 J=1,11
0081            L=6
0082            DO 10 I=1,120
0083            PMATR(I)=BLANK
0084      10    CONTINUE
0085            DO 12 K=1,19
0086            IF(CNT.EQ.'1') GO TO 6
0087            IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'2') GO TO 7
0088            GO TO 5
0089      6     IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'3') GO TO 7
0090            GO TO 5
0091      7      PMATR(L)=LABLED(K)
0092            PMATR(L+1)=SLASH
0093            PMATR(L+2)=CLASFY(2*K)
0094            IF(CLASFY(2*K-1).EQ.' ') GO TO 5
0095             PMATR(L+2)=CLASFY(2*K-1)
0096             PMATR(L+3)=CLASFY(2*K)
0097        5   CONTINUE
0098            K=K+1
0099            L=L+6
0100       12   CONTINUE
0101            WRITE(6,99) NLINE,(PMATR(I),I=6,120)
0102       99   FORMAT(1H0,2X,I3.2X,115A1)
0103            NLINE=NLINE+10
0104       98   CONTINUE
0105            CNT=TWO
0106      97     CONTINUE
```

```
0107            WRITE(6,100)
0108            WRITE(6,101)
0109            WRITE(6,102) (PARAL(I),I=1,5)
0110           WRITE(6,126)
0111            WRITE(6,103) (PARPP(I),I=1,12)
0112            WRITE(6,104) (PARSCA(I),I=1,12)
0113            WRITE(6,105) (PARCP(I),I=1,5),PARCPR(12)
0114            WRITE(6,106) (PARVAR(I),I=1,5),PARVAR(12)
0115            WRITE(6,107) (PARVARS(I),I=1,5),PARVARS(12)
0116           WRITE(6,132)
0117            CALL BNT
0118            WRITE(6,108)
0119           WRITE(6,130) PCC1,PCC2
0120            WRITE(6,109)
0121           WRITE(6,131) PCCS1,PCCS2
0122            WRITE(6,110) (PARCLAB(I),I=1,8)
0123            WRITE(6,111)
0124            WRITE(6,136) (CLUD(I),I=1,2)
0125            DO 96 J=1,PARCATS
0126            WRITE(6,112) PARLABL(J),(PARCAT(J,K),K=1,PARCATS)
0127    96      CONTINUE
0128           WRITE(6,132)
0129            CALL BNT
0130            CALL BNT
0131            WRITE(6,113)
0132           DO 50 K=1,BARCATS
0133           WRITE(6,134) BARLABL(K),BARCLAB(LL),RCVA(K,LL),LL=1,BARCATS)
0134    50     CONTINUE
0135           WRITE(6,135) BAT,*
0136    202       IF(CLUDUM.EQ.1) RETURN
0137           WRITE(6,129)
0138           CNT=BASE
0139           DO 297 JJ=1,2
0140           K=1
0141           WRITE=10
0142           WRITE(6,123) CNT
0143           WRITE(6,124)
0144           WRITE(6,125) (CHL(I),I=1,19)
0145           DO 298 J=1,11
0146           L=6
0147           DO 210 I=1,120
0148           PMATR(I)=BLANK
0149    210     CONTINUE
0150           DO 212 N=1,19
0151           IF(CNT.EQ.'1') GO TO 206
0152           IF(TYPE(K).EQ.'0'.OR.TYPE(K).EQ.'2') GO TO 207
0153           GO TO 205
0154    206       IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'3') GO TO 207
0155           GO TO 205
0156    207     PMATR(L)=LABLED(K)
0157           PMATR(L+1)=SLASH
0158           PMATR(L+2)=CLOGT(2*K)
0159           IF(CLOGT(2*K-1).EQ.' ') GO TO 5
0160            PMATR(L+2)=CLOGT(2*K-1)
0161            PMATR(L+3)=CLOGT(2*K)
0162      205   CONTINUE
```

```
0163                K=K+1
0164                L=L+6
0165     212        CONTINUE
0166                WRITE(6,99) NLINE,(PMATR(I),I=6,129)
0167                NLINE=NLINE+10
0168       299      CONTINUE
0169                CNT=7.0
0170     297        CONTINUE
0171                WRITE(6,128)
0172                WRITE(6,101)
0173                WRITE(6,102) (RUCAPL(I),I=1,8)
0174                WRITE(6,127)
0175                WRITE(6,103) (RUPPR(I),I=1,12)
0176                WRITE(6,104) (RUNCR(I),I=1,12)
0177                WRITE(6,105) (RUCIR(I),I=1,6),RUCPR(12)
0178                WRITE(6,106) (RUMAR(I),I=1,6),RUVAR(12)
0179                WRITE(6,107) (RUPRS(I),I=1,6),RURANS(12)
0180                WRITE(6,172)
0181                CALL BNT
0182                WRITE(6,108)
0183                WRITE(6,130) PCL1,PCL2
0184                WRITE(6,109)
0185                WRITE(6,131) PCLG1,PCLG2
0186                WRITE(6,110) (RUCLAR(I),I=1,8)
0187                WRITE(6,111)
0188                WRITE(6,136) (FLAG(I),I=1,2)
0189                DO 2" J=1,RUCATS
0190                WRITE (6,112) RULARL(J),(EUCAIN(J,K),K=1,BUCATS)
0191       2        CONTINUE
0192                WRITE(6,172)
0193                CALL BNT
0194                CALL BNT
0195                WRITE(6,113)
0196                DO 250 K=1,RUCATS
0197                WRITE(6,134) (BUCLAR(K),BUCLA3(LL),RCVU(K,LL),LL=1,BACATS)
0198     250        CONTINUE
0199                WRITE(6,135) RUTOT
0200                RETURN
0201                END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	006000	1536	RW,I,CON,LCL
2	$PDATA	000006	3	RW,D,CON,LCL
3	$IDATA	001714	486	RW,D,CON,LCL
4	$VARS	000462	153	RW,D,LCN,LCL
6	CBIAS	005122	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLCOM	000642	209	RW,D,OVR,GBL
9	SEG	000004	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BIAPRT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BACATS	I*2	6-003314	BATST	I*2	6-003766	BLANK	L*1	4-000170	BUCATS	I*2	6-001556	BUTST	I*2	6-003770
CLADUM	I*2	7-000000	CLODUM	I*2	7-000002	CNT	I*2	4-000450	DN	I*2	6-001560	I	I*2	4-001562
I	I*2	4-000436	J	I*2	4-000440	JJ	I*2	4-000452	K	I*2	4-000442	L	I*2	4-000444
LL	I*2	4-000460	M	I*2	4-000446	N	I*2	4-000456	NLINE	I*2	4-000454	Z F	I*2	4-000430
PASS	I*2	F-000006*	PCCG1	R*4	6-003336	PCCG2	R*4	6-003342	PCG1	R*4	6-003326	PCG2	R*4	6-003332
PCLG1	R*4	6-003356	PCLG2	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	SLASH	I*2	4-000432
TAPN0	I*2	F-000004*	C	I*2	6-001554	TN3	I*2	4-000434						

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
BACAIN	R*4	6-000220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BAC2R	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAP2P	I*2	6-000000	000030	12	(12)
BARANS	R*4	6-001360	000060	24	(12)
BAUNC2	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVU	I*2	6-003566	000200	64	(8,8)
BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001550	000010	4	(8)
BUC2R	I*2	6-000170	000030	12	(12)
BULABL	L*1	6-001570	000010	4	(8)
BUP2P	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNC2	I*2	6-000110	000030	12	(12)

BIAPRT.FTN /TR:BLOCKS/WR

BVAR	R*4	6-001330	000060	24	(12)
COORS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	000642	209	(416)
CLOUT	L*1	6-000000	000642	209	(416)
CLUD	R*4	6-004052	000010	4	(2)
COL	I*2	4-000352	000046	19	(19)
IBUF	L*1	F-000[illegible]	000001	1	(1)
LABLED	L*1	6-002773	000321	104	(209)
PB	L*1	4-000000	000170	60	(120)
PVATR	L*1	4-000171	000170	60	(120)
ROWS	R*4	6-004112	000000	256	(2,8,8)
SEGNO	L*1	9-000000	000004	2	(4)
TOWNS	R*4	6-004012	000100	32	(2,8)
TYPE	L*1	6-002452	000321	104	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000054	2	1-000430	3	**	5	1-[illegible]	[illegible]	1-001214
7	1-001254	10	**	12	**	20	**	[illegible]	**
96	**	97	**	98	**	[illegible]	[illegible]	[illegible]	[illegible]
101'	3-000054	102'	3-000074	103'	3-000112	[illegible]	[illegible]	[illegible]	[illegible]
106'	3-000250	107'	3-000314	108'	3-000360	[illegible]	[illegible]	[illegible]	[illegible]
111'	3-000470	112'	3-000514	113'	3-000564	[illegible]	**	[illegible]	[illegible]
116'	3-000654	117'	3-000730	118'	3-000762	[illegible]	[illegible]	[illegible]	[illegible]
121'	**	122'	3-001114	123'	3-001164	[illegible]	[illegible]	[illegible]	[illegible]
126'	3-001300	127'	3-001336	128'	3-001374	[illegible]	[illegible]	[illegible]	[illegible]
131'	3-001510	132'	3-001610	133'	**	[illegible]	[illegible]	[illegible]	[illegible]
136'	3-001550	202	1-003310	205	1-004030	[illegible]	[illegible]	[illegible]	[illegible]
210	**	212	**	250	**	[illegible]	**	[illegible]	**

FUNCTIONS AND SUBROUTINES REFERENCED

BNT

TOTAL SPACE ALLOCATED = 016400 3712

NO FPP INSTRUCTIONS GENERATED

BIAPRT,LP:=BIAPRT

```
*FORTRAN IV-PLUS V02-51              08105155    09-MAR-78          PAGE 1
CORED.FTN       /TR:BLOCKS/WR
0001            SUBROUTINE CDPED(IBUF,R,FILE)
0002            IMPLICIT INTEGER(A-Z)
0003            INTEGER*2 ISTAT(2),IPRM(6)
0004             EQUIVALENCE(ISTAT(1),IB)
0005             BYTE IB(2)
0006            LOGICAL*1 IBUF(1)
0007            DIMENSION IHDAT(1)
0008            DIMENSION IA(37)
0009            DATA XDEV/2HXT/
0010            DATA MDEV/2HMT/
0011            RECORD=R
0012            IF(RECORD.GT.0)  GO TO 10
0013             ERCNT=0
0014             IF(RECORD.EQ.-1)  GO TO 6
0015    12      CONTINUE
0016            FILE=C
        D       WRITE(5,49)                                          IOERR**
        CTYPE 49
0017    49      FORMAT(10X,' TYPE M OR X FOR TAPE DEVICE CODE'/)
0018           READ(1,51,END=99)  IA
        C       ACCEPT 51,IA
0019            IF(IA(1).EQ.'M')  GO TO 13
0020            IF(IA(1).EQ.'X')  GO TO 14
0021    99      CONTINUE
0022            WRITE(6,97)
0023    97      FORMAT(10X,' CARD ERROR  '//)
0024            WRITE(6,97)
0025            STOP
0026    13      IDEV=MDEV
0027            GO TO 15
0028    14      IDEV=XDEV
0029    15      CONTINUE
        D       WRITE(5,50)                                          IOERR**
        C       TYPE 50
0030    50      FORMAT(10X,' TYPE TAPE UNIT NUMBER = 0 OR 1')
0031            READ(1,51,END=99)  IA
        C       ACCEPT 51,IA
0032    51      FORMAT(37A2)
0033             IF(IA(1).EQ.'0')  GO TO 18
0034             IF(IA(1).EQ.'1')  GO TO 19
0035             GO TO 99
0036    18       IUNT=0
0037             GO TO 20
0038    19       IUNT=1
0039    20       CONTINUE
0040            ILUN=9
0041            IDS=0
0042            ISR=0
0043            CALL ASNLUN(ILUN,IDEV,IUNT,IDS)
0044            IF(IDS.LT.0)  GO TO 1
0045            CALL GETADR(IPRM,IBUF)
0046    6        CONTINUE
0047            CALL QIO(%2400,ILUN,1,,ISTAT,IPRM,ISR)
0048            IF(ISR.LT.0)  GO TO 2
0049             IF(RECORD.EQ.-1)  GO TO 11
0050            IPRM(2)=3600
```

```
0051     10     CONTINUE
0052            ISW=0
0053            RECORD=RECORD+1
0054            CALL QIO("1000,ILUN,1,,ISTAT,IPRM,ISW)
0055            IF(ISW.LT.0)  GO TO 3
0056            CALL WAITFR(1,IDS)
0057            IF(IDS.LT.0)  GO TO 4
        C        WRITE(6,101) (IB(I),I=1,2),ISTAT(2)
0058     101    FORMAT(1H0,10X,2I4,3X,I5,2X,'BYTES TRANSFERED')
0059            ERCODE=IB(1)
0060            IF(ERCODE.LT.0)  GO TO 5
0061            FF=0
0062            DO 800 JJ=1,720
0063     798    IF(IBUF(JJ).LT.0) GO TO 799
0064     800    CONTINUE
0065            RETURN
0066     799    IBUF(JJ)='1'
0067            IF(FF.EQ.1) GO TO 798
0068            FF=1
0069            WRITE(6,1000)
0070     1000   FORMAT(1H '***** THIS SEGMENT CONTAINS BAD DATA *****'
0071            GO TO 798
0072     1      CONTINUE
0073            WRITE(6,100)  IDS
0074     100    FORMAT(1H ,'  ASLUN CALL DSW=  ',I6)
0075            STOP
0076     2      CONTINUE
0077            WRITE(6,200) ISP
0078     200    FORMAT(1H ,'REWIND DSW = ',I6)
0079            STOP
0080     3      CONTINUE
0081            WRITE(6,300)  ISW
0082     300    FORMAT(1H ,'  READ QIO DSW = ',I6)
0083            STOP
0084     4      CONTINUE
0085            WRITE(6,400)  IDS
0086     400    FORMAT(1H ,'  WAIT DSW = ',I6)
0087            STOP
0088     5      CONTINUE
0089            IF(ERCODE.EQ.-10)  FILE=FILE+1
0090            IF(ERCODE.EQ.-10)  RETURN
0091             IF(ERCODE.NE.-4)  GO TO 7
0092             ERCNT=ERCNT+1
0093             RETURN
0094     11     CONTINUE
0095            WRITE(6,700)
0096     700    FORMAT('  ')
0097            IF(ERCNT.EQ.0)  RETURN
0098            WRITE(6,600)  ERCNT
0099     600    FORMAT('1',10X,' TAPE ERRORS ENCOUNTERED = ',I5)
0100            RETURN
0101     7      CONTINUE
0102            WRITE(6,500)  ERCODE
0103     500    FORMAT(1H0,'  I/O STATUS BLOCK ERROR CODE= ',I6)
0104            STOP
0105            END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001420	392	RW,I,CON,LCL
2	$PDATA	000014	6	RW,D,CON,LCL
3	$IDATA	000472	157	RW,D,CON,LCL
4	$VARS	000164	59	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CORED		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ERCNT	I*2	4-000142	ERCODE	I*2	4-000160	FF	I*2	4-000162	FILE	I*2	F-000000*	IDS	I*2	4-000144
IDS	I*2	4-000152	ILUN	I*2	4-000151	ISR	I*2	4-000154	IUN	I*2	4-000165	IUNIT	I*2	4-000146
JJ	I*2	4-000164	MREV	I*2	4-000136	R	I*2	F-000004*	RECORD	I*2	4-000140	XTN	I*2	4-000134

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
IA	I*2	4-000022	000112	37	(37)
IB	L*1	4-000000	000002	1	(2)
IBUF	L*1	F-000002*	000001	0	(1)
IHDAT	I*2	4-000020	000002	1	(1)
IPRM	I*2	4-000004	000014	6	(6)
ISTAT	I*2	4-000000	000004	2	(2)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000774	2	1-001340	3	1-001104	4	1-001150	5	1-001214
6	1-000472	7	1-001354	10	1-000544	11	1-001264	12	**
13	1-000232	14	1-000250	15	1-000264	16	1-001352	17	1-000364
20	1-000402	49'	**	50'	**	51'	3-000024	97'	3-000000
99	1-000156	100'	3-000110	101'	**	200'	3-000142	300'	3-000166
400'	3-000216	500'	3-000314	600'	3-000250	700'	3-001240	798	1-000452
799	1-000712	800	**	1000'	3-000030				

FUNCTIONS AND SUBROUTINES REFERENCED

ASNLUN GETADR OIP WAITFR

CORED.FTN /TR:BLOCKS/NA
TOTAL SPACE ALLOCATED = 002314 614
@
NO FPP INSTRUCTIONS GENERATED
@
CORED,LP:=CORED

```
SEGTRAN IV-PLUS V02-51                    CE:C6[C7    09-MAR-78          PAGE 1
C:\RED.FTN        /TR:I-LOCKS/:5
0001              SUBROUTINE CHARGE
0002              IMPLICIT INTEGER(A-Z)
0003              INCLUDE 'COMMON1.DAT'
0004  *           BYTE CLASFY(416),PLTYPE(416)
0005  *           LOGICAL*1 TYPE(2,5),LABLE(205)
0006  *           LOGICAL*1 RALBL(7),FACLAB(5),SUCLAB(8),PUCLAB(8)
0007  *           REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
0008  *           REAL CLAD,CLUD,TCONS,RCONS,CCONS
0009  *           DIMENSION CLAT(2),CLUB(2),TCONS(2,8),CCONS(2),RCONS(2,5,6)
0010  *           DIMENSION RCVA(5,7),SCVU(5,6)
0011  *           DIMENSION PARAM(42),SUBP(12)
0012  *           COMMON/CRIAS/PAR P,SUPCP,BALANCE,AUDANCE,SACCE,SUCCESS,BACAT ,PUCAT ,
      *                  PAYAB,CUMAR,CARANS,ERRANS,A,PE,IN,TC,SUCETS,PCLAB,
      *                  SUBALL,SALABL,CLASFY,TYPE,LABLE,PACATS,PACLAB,
      *                  PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
      *                  RCVA,SCVU,DATET,BLTZT,CLAD,CLUB,TCONS,RCONS,CCONS
0013  *           COMMON/DUMMY/CLADUM,CLUBUM
0014  *           COMMON/CLOUT/CLOUT
0015  *           REAL A(3,2)
0016  *           DIMENSION BANDP,(12),BANDC(12),PACE#(12),PUCAT(12)
0017  *           REAL BACAT(6,5),BUCAT(5,6,BALL,R(12),CUMAR(12),SARANS(12),CLSANS(12)
0018              IU=1
0019              DO 10 I=1,2
0020              READ(IU,101) CCONS(I)
0021        101   FORMAT(F10.5)
0022              READ(IU,102) (TCONS(I,J),J=1,8)
0023              DO 9 J=1,8
0024              READ(IU,100) (RCONS(I,J,K),K=1,6)
0025        9     CONTINUE
0026        10    CONTINUE
0027        102   FORMAT(8F10.5)
0028              RETURN
0029              END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	SCODE1	000334	110	RW,I,CON,LCL
3	BLANK	000010	5	RW,D,CON,LCL
4	SVA=5	000010	4	RW,D,CON,LCL
6	CBLAS	005120	1321	RW,D,ZVR,GBL
7	DUMMY	000004	2	RW,D,ZVR,GBL
8	CLCOM	000642	209	RW,D,ZVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CONRED		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BACATS	I*2	6-003316	BATOT	I*2	6-003766	BUCATS	I*2	6-001556	BUTOT	I*2	6-003720	CLADUM	I*2	7-000000
CLUDUM	I*2	7-000002	DF	I*2	6-001550	DJ	I*2	6-001552	I	I*2	6-001542	J	I*2	6-000000
J	I*2	4-000004	K	I*2	4-000006	PCOG1	R*4	6-003336	PCOG2	R*4	6-003340	PCOS	R*4	6-003326
PCO2	R*4	6-003332	PCLG1	R*4	6-003356	PCLG2	R*4	6-003342	PCL1	R*4	6-003346	PCL2	R*4	6-003352
TC	I*2	6-001554												

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
BACAIN	R*4	6-000220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACOR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPOP	I*2	6-000000	000030	12	(12)
BARANS	R*4	6-001350	000060	24	(12)
BAUNCJ	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVU	I*2	6-003566	000200	64	(8,8)
BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUCOR	I*2	6-000170	000030	12	(12)
BULABL	L*1	6-001570	000010	4	(8)
BUPOP	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNCJ	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001300	000060	24	(12)
CORES	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	000642	209	(418)

```
FORTRAN IV-PLUS V02-51          R5:05:07    05-MAR-78          PAGE 3
CONRED.FTN      /TR:PL2C:S/LP
 CLOSE  Loc  0-000000  000542   279    (416)
 CLOC   Rel  0-004052  000010     4    (2)
 LBLRD  Loc  0-002773  000321   209    (209)
 RCOLS  Rel  0-004112  001000   256    (2,6,6)
 TCOLS  Rel  0-004012  000100    32    (2,6)
 TYPE   Loc  0-002452  000321   209    (209)
 .
 .
LABELS
 .
 LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS
 .
 9         **            10        **          100*    3-000004        101*    3-000007
 .
 .
TOTAL SPACE ALLOCATED = 006346  1651
 .
NO FPP INSTRUCTIONS GENERATED
 .
CONRED,LP:=CONRED
```

```
MFORTRAN IV-PLUS V02-51                    08:10:13    09-MAR-78          PAGE 1
MDTTL.FTN       /TRIBL7CKS/NR
0001            SUBROUTINE MDTTL(PTL)
0002            IMPLICIT INTEGER*2(A-Z)
0003            LOGICAL*1  MSD(12),PTL(1)
0004            DATA MSD/1HM,1HE,1HT,1HN,1H ,1H ,1HS,1HT,1H ,1HD,1HE,1HL/
0005            DO 1  I=1,12
0006     1      PTL(I)=MSD(I)
0007            RETURN
0008            END
```

```
FORTRAN IV-PLUS V02-51          [illegible]    09-MAR-79          PAGE 2
MDTTL.FTN       /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER   NAME      SIZE                ATTRIBUTES

  1      $CODE1  000102    33          RW,I,CON,LCL
  3      $IDATA  000012     5          RW,D,CON,LCL
  4      $VARS   000016     7          RW,D,CON,LCL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 MDTTL        1-000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 I      I*2   4-000014

ARRAYS

 NAME   TYPE  ADDRESS      SIZE      DIMENSIONS

 MSG    L*1   4-000000  000014    6   (12)
 PTL    L*1   F-000002* 000001    0   (1)

LABELS

 LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS

 1        **

TOTAL SPACE ALLOCATED = 000132    45

NO FPP INSTRUCTIONS GENERATED

MDTTL.LP;*MDTTL
```

```
 #F2RTRAN IV-PLUS V02-51                  18106116   29-MAR-78          PAGE 1
 FNAMF,FTN       /*:-L*C*S/*R
 0001            SUBROUTINE FNAMF(F,,PN)
 0002            IMPLICIT INTEGER(A-Z)
 0003            LOGICAL*1  F(9),PN(*),FIELD(9)
 0004            DATA FIELD/1HF,1H-,1HF,1HC,1HD,1H /
 0005            DO 11  I=1,5
 0006            PN(I)=FIELD(I)
 0007            PN(I+5)=F(I)
 0008      11    CONTINUE
 0009            RETURN
 0010            END
```

```
FORTRAN IV-PLUS V02-54          [illegible]  15-MAR-77          PAGE 2
FNAME.FTN       /TR:BLOCKS/WD

PROGRAM SECTIONS

NUMBER   NAME      SIZE          ATTRIBUTES
  1      $CODE1   000132    45   RW,I,CON,LCL
  3      $IDATA   000024    10   RW,D,CON,LCL
  4      $VARS    000010     4   RW,D,CON,LCL


ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 FNAME        1-000000


VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 II     I*2   4-000016


ARRAYS

 NAME   TYPE  ADDRESS      SIZE      DIMENSIONS

 FIELD  L*1   4-000000  000006    3   (6)
 FLD    L*1   F-000002* 000001    0   (1)
 PN     L*1   F-000004* 000001    0   (1)


LABELS

 LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS

 11        **


TOTAL SPACE ALLOCATED = 000166    59

NO FPP INSTRUCTIONS GENERATED

FNAME,LP1=FNAME
```

```
@FORTRAN IV-PLUS V02-51          _CE1C6119     29-MAR-79          PAGE 1
KNT,FTN          /TR:BLOCKS/WR
0001          SUBROUTINE KNT
0002          IMPLICIT INTEGER(A-Z)
0003          COMMON/PCNT/LINE,PAGE
0004          LINE=LINE+1
0005          IF(LINE .LT. 60) GO TO 10
0006          PAGE=PAGE+1
0007          LINE=0
0008          WRITE(6,100) PAGE
0009       10 RETURN
0010      100 FORMAT('1',26X,'CAS/CAS INTERFACE TAPE PRINTOUT',14X,'PAGE',I4/)
0011          END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	$CODE1	000076	31	PIC,I,LCL,LCL
3	$IDATA	000052	25	PIC,D,LCL,LCL
6	PDT	000004	2	PIC,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
KNT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LINE	I*2	6-000000	PAGE	I*2	6-000002									

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000074	100'	3-000000						

TOTAL SPACE ALLOCATED = 000164 58

NO FPP INSTRUCTIONS GENERATED

KNT,LP:=KNT

```
0FORTRAN IV-PLUS V02-51                    02106122    09-MAR-78          PAGE 1
MEAN.FTN          /TR:3L3C/S/H2
0001              SUBROUTINE MEAN(FLD,PM,NCH,FLG)
0002              IMPLICIT INTEGER*2(A-Z)
0003              LOGICAL*1 FLD(1),PM(1),FLG(1),PT
0004               LOGICAL*1 BLK
0005              DATA PT/'+'/
0006               DATA BLK/' '/
0007              NFF=11
0008              IF(FLG(1).EQ.'F')  NFF=12
0009              K=1
0010               DO  3  J=1,NCH
0011              NE=(J-1)*132
0012               PM(K+1)=FLD(NFF+1)
0013               PM(K+2)=FLD(NFF+2)
0014               IF(PM(K+1).NE.'0')  GO TO 10
0015               PM(K+1)=BLK
0016               IF(PM(K+2).EQ.'0')  PM(K+2)=BLK
0017    10         PM(K+3)=FLD(NFF+3)
0018               PM(K+4)=PT
0019               PM(K+5)=FLD(NFF+4)
0020               PM(K+6)=FLD(NFF+5)
0021               PM(K+7)=BLK
0022               PM(K+8)=FLD(NFF+6)
0023               PM(K+9)=FLD(NFF+7)
0024               IF(PM(K+8).NE.'0')  GO TO 11
0025               PM(K+8)=BLK
0026               IF(PM(K+9).EQ.'0')  PM(K+9)=BLK
0027    11         PM(K+10)=FLD(NFF+8)
0028               PM(K+11)=PT
0029               PM(K+12)=FLD(NFF+9)
0030               NFF=NFF+9
0031    3          CONTINUE
0032               RETURN
0033               END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000514	166	RW,I,CON,LCL
3	$DATA	000036	15	RW,D,CON,LCL
4	.VARS	000012	5	RW,D,OVR,LCL
5	$TEMPS	000002	1	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
MEAN		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BLK	I*2	4-000000	J	I*2	4-000036	K	I*2	4-000034	L	I*2	4-000032	M	I*2	5-000000
ZZZ	I*2	4-000012	PI	R*4	4-000030									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
FLG	L*1	5-000002*	000001	0	(1)
FLG	L*1	5-000010*	000001	0	(1)
PM	L*1	5-000004*	000001	0	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
3	**	10	1-000260	11	1-000427				

TOTAL SPACE ALLOCATED = 000546 187

NO FPP INSTRUCTIONS GENERATED

MEAN,LP:=MEAN

```
@F2RTRAN IV-PLUS V02-54                    08105125    09-MAR-75          PAGE 1
P2P.FTN          /TR:PROCS/WR
0001             SUBROUTINE P2P(FLG,PP,FLG)
0002             IMPLICIT INTEGER(A-Z)
0003             LOGICAL*1 FLG(1),PP(2),PAR(2),FLG(1)
0004             DATA PAR/2H(,2H)/
0005             BFF=6
0006             IF(FLG(1).EQ.'F')  BFF=5
0007             PP(3)=PAR(1)
0008             DO  1  I=1,5
0009             PP(I+3)=FLG(I+BFF)
0010     1       CONTINUE
0011             PP(9)=PAR(2)
0012             RETURN
0013             END
```

```
FORTRAN IV-PLUS V02-51          [illegible]     09-MAR-73          PAGE 2
P2P.FTN          /TS:P2P/OBJ/NOP
PROGRAM SECTIONS

  NUMBER   NAME      SIZE                  ATTRIBUTES

    1      $CODE1   000216    71          RW,I,CON,LCL
    3      $DATA    000036    15          RW,D,CON,LCL
    4      $VARS    000016     3          RW,D,OVR,LCL

ENTRY POINTS

  NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

  P2P           1#000000

VARIABLES

  NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

  I      I*2   4#000004   CPP    I*2   4#000002

ARRAYS

  NAME   TYPE  ADDRESS      SIZE      DIMENSIONS

  FLD    L*1   F#000002* 000001    0    (1)
  FLG    L*1   F#000056* 000001    0    (1)
  PAR    L*1   4#000000  000002    1    (2)
  PP     L*1   F#000054* 000001    0    (1)

LABELS

  LABEL   ADDRESS      LABEL   ADDRESS      LABEL   ADDRESS      LABEL   ADDRESS      LABEL   ADDRESS

  1         **

TOTAL SPACE ALLOCATED = 000262    89

NO FPP INSTRUCTIONS GENERATED

P2P,LP:=P2P
```

```
FTN5,L.                                                                    PAGE 1
SNAME,FTN       /TR:P,OPS/A,R
0001            SUBROUTINE SNAME(FLG,PN,FLG)
0002            IMPLICIT INTEGER(A-Z)
0003            LOGICAL*1  FLG(1),PN(1),SUBCL(6),FLG(1)
0004            DATA SUBCL/'NS','ME',' ','HR','RC',1HL,1H /
0005            FFF=0
0006            IF(FLG(1).EQ.'FF') FFF=7
0007            DO 11  II=1,6
0008            P (II)=SUBCL(II)
0009            P (II+6)=FLG(II+FFF)
0010     11     CONTINUE
0011            RETURN
0012            END
```

```
FORTRAN IV-PLUS V02-51          [illegible]    09-MAR-78          PAGE 2
SNAME.FTN       /TRACEBACKS/WR

PROGRAM SECTIONS

NUMBER   NAME        SIZE                  ATTRIBUTES

  1      $CODE1  000252    85          RW,I,CON,LCL
  3      $DATA   000036    15          RW,D,CON,LCL
  4      $VARS   000012     5          RW,D,CON,LCL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 SNAME        1-000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 II     I*2   4-000010   OFF    I*2   4-000012

ARRAYS

 NAME   TYPE  ADDRESS        SIZE        DIMENSIONS

 FLD    L*1   F-000002* 000001     3     (1)
 FLG    L*1   F-000005* 000001     3     (1)
 PN     L*1   F-000004* 000001     3     (1)
 SUBCL  L*1   4-000000  000006     3     (6)

LABELS

 LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS

 11        **

TOTAL SPACE ALLOCATED = 000252    85

NO FPP INSTRUCTIONS GENERATED

SNAME,LP:=SNAME
```

```
$FORTRAN IV-PLUS V02-51                    UB106137     09-MAR-78          PAGE 1
STAT,FTN        /TR:FLPOC(S/NP
0001            SUBROUTINE STAT(IOF)
0002            IMPLICIT INTEGER*2(A-Z)
0003            LOGICAL*1 IOF(1)
0004            COMMON /PCNT/LINE,PAGE
0005            LINE=66
0006            CALL EXIT
0007            IF(IOF(3),EQ,'F')  GO TO 8
0008            WRITE(6,200)
0009    200     FORMAT(5X,'SUBCLASS STATISTICS RECORD')
0010            GO TO 9
0011    8       WRITE(6,201)
0012    201     FORMAT(5X 'FIELD STATISTICS RECORD')
0013    9       CONTINUE
0014            WRITE(6,202)  (IOF(I),I=3,4)
0015    202     FORMAT(5X,'RESPONSE SEQ. NO.  ',2A1)
0016            WRITE(6,203)  (IOF(I),I=5,7)
0017    203     FORMAT(5X,'#  SUBCLASSES  '3A1)
0018            WRITE(6,204)  (IOF(I),I=8,9)
0019    204     FORMAT(5X,'#  SUBCLASS SETS ',2A1)
0020            WRITE(6,205)  (IOF(I),I=10,11)
0021    205     FORMAT(5X,'#  OF CHANGES    ',2A1)
0022            LINE=5
0023            RETURN
0024            END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001467	152	RW,I,LCL,LCL
3	$IDATA	000256	87	RW,I,LCL,LCL
4	$VARS	000002	1	RW,D,GBL,LCL
5	PDOT	000004	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
STOMP		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LINE	I*2	4-000000	N	I*2	4-000000	PAGE	I*2	4-000002						

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
IBUF	L*1	F-000002*	000001	0 (1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
8	1-000120	9	1-000146	200'	3-000000	201'	3-000040	202'	3-000074
203'	3-000126	204'	3-000156	205'	3-000210				

FUNCTIONS AND SUBROUTINES REFERENCED

KNT

TOTAL SPACE ALLOCATED = 000744 242

52 PPP INSTRUCTIONS GENERATED

STOMP,LP:=STOMP

```
FORTRAN IV-PLUS V02-51                  08:05:42    09-MAR-76          PAGE 1
PRT.FTN          /TRACEBACKS/WP
0001              SUBROUTINE PRT
0002              IMPLICIT INTEGER(A-Z)
0003              COMMON/PCNT/LINE,PAGE
0004              LINE=LINE+1
0005              IF(LINE .LT. 56) GO TO 10
0006              PAGE=PAGE+1
0007              LINE=0
0008              WRITE(3,100)
0009          10  RETURN
0010         100  FORMAT('1',71X,'    ')
0011              END
```

BNT,FTN /TR:BLOCKS/RE

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	$CODE1	000066	27	RW,I,CON,LCL
3	$IDATA	000014	6	RW,D,CON,LCL
6	PCNT	000004	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BNT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LINE	I*2	6-000000	PAGE	I*2	6-000002									

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000064	100'	3-000000						

TOTAL SPACE ALLOCATED = 000106 35

0 FPP INSTRUCTIONS GENERATED

BNT,LP:=BNT

```
*FORTRAN IV-PLUS V02-51                    18:06:46    29-MAR-79          PAGE 1
CPIP2.FTN          /TR:HL2CRS/.R
0001               SUBROUTINE CPIP2(FLD,CLASS,P1,P2)
0002               IMPLICIT INTEGER*2(A-Z)
0003               LOGICAL*1  FLD(1),CLASS(1)
0004               DO  2  I=1,4
0005               CLASS(I)=FLD(I)
0006     2         CONTINUE
0007               DECODE(5,100,FLD(3))  P1
0008     100       FORMAT(I5)
0009               DECODE(5,100,FLD(5))  P2
0010               RETURN
0011               END
```

```
FORTRAN V=PLUS /OP=51                    OS-VAR-74          PAGE 2
CRIP2,FTN       /TR:BLOCKS/NO
PROGRAM SECTIONS

NUMBER   NAME      SIZE              ATTRIBUTES

  1      BDDDF1  000224    74        R=,I=,CON,LCL
  3      FIMATA  000030    12        R=,D=,CON,LCL
  4      EVARS   000002     1        R=,D=,CON,LCL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 CRIP2          1=000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRES

 I      I*2   4=000000   P1     I*2   F=000006*  P2     I*2   F=000010*

ARRAYS

 NAME   TYPE  ADDRESS        SIZE        DIMENSIONS

 CLASS  L*1   F=000024* 000001      0     (1)
 FLD    L*1   F=000002* 000031      0     (1)

LABELS

 LABEL   ADDRESS          LABEL   ADDRESS          LABEL   ADDRESS          LABEL   ADDRESS          LABEL   ADDRESS

 2          **            101*    3=000010

TOTAL SPACE ALLOCATED = 000256    87

92 FPP INSTRUCTIONS GENERATED

CRIP2,LP:=CRIP2
```

```
$FORTRAN IV-PLUS V02-51                  08:05:51    09-MAR-78          PAGE 1
MV.FTN                  /TR:BLOCKS/WP
0001                    SUBROUTINE MV(FLD,PB,NC)
0002                    IMPLICIT INTEGER(A-Z)
0003                    LOGICAL*1 FLD(1),PB(1)
0004                    DO  8  J=1,NC
0005     8              PB(J)=FLD(J)
0006                    RETURN
0007                    END
```

MV.FTN /TR:SLBCKS/AS

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	$CODE1	000124	42	RW,I,CON,LCL
3	$IDATA	000024	10	RW,D,CON,LCL
4	$VARS	000002	1	RW,D,OVR,LCL
5	$TEMPS	000002	1	RW,D,OVR,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
MV		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRES
J	I*2	4-000000	N	I*2	F-000000*									

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
FLD	L*1	F-000002*	000001	3	(1)
PR	L*1	F-000003*	000001	3	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
8	**								

TOTAL SPACE ALLOCATED = 000154 54

NO FPP INSTRUCTIONS GENERATED

APPENDIX B

The values Dw and Ds are computed as follows:

$Dw = C_1 + \sum_{p=1}^{8} \sum_{q=1}^{8} RpqWpWq + \sum_{p=1}^{8} TpWp$ or 0 if Category W not used

$Ds = C_2 + \sum_{p=1}^{8} \sum_{q=1}^{8} UpqSpSq + \sum_{p=1}^{8} VpSp$ or 0 if Category S not used

where C_1, C_2, $\{Rpq\}_{p=1}^{8}{}_{q=1}^{8}$, $\{Tp\}_{p=1}^{8}$, $\{Upq\}_{p=1}^{8}{}_{q=1}^{8}$, and $\{Vp\}_{p=1}^{8}$ are constants, supplied by user

W_1 = Bias corrected estimate for Category W

S_1 = Bias corrected estimate for Category S

W_2 = Machine estimate for Category W

S_2 = Machine estimate for Category S

W_3 = Random estimate for Category W

S_3 = Random estimate for Category S

W_4 = Variance of Bias corrected estimate of W

S_4 = Variance of Bias corrected estimate of S

$$W_5 = \frac{W:W}{W:W + S:W + N:W}$$

$$S_5 = \frac{S:S}{W:S + S:S + N:S}$$

$W_6 = S_6$ = PCC1

$$W_7 = S_7 = \frac{N:N}{W:N + S:N + N:N}$$

$W_8 = S_8$ = PCC2

APPENDIX C

a. Constants for Dw calculations

Card 1 contains C_1 as follows:
blanks or XXX.XXXXX in first 10 columns only

Card 2 contains T_1 thru T_8 as follows:
blanks or XXX.XXXXX for each entry. A maximum of 10 columns wide starting in columns 1, 11, 21, etc.

Card 3 thru 10 contain $R_{1,1}$ to $R_{8,8}$. $R_{1,1}$ thru $R_{1,8}$ on card 3, $R_{2,1}$ thru $R_{2,8}$ on card 4, etc. Format same as card 2.

b. Constants for Ds calculations

Card 1 contains C_2 as follows:
blanks or XXX.XXXXX in first 10 columns only

Card 2 contains V_1 thru V_8 as follows:
blanks or XXX.XXXXX for each entry. A maximum of 10 columns wide starting in columns 1, 11, 21, etc.

Card 3 thru 10 contain $U_{1,1}$ to $U_{8,8}$. $U_{1,1}$ thru $U_{1,8}$ on card 3, $U_{2,1}$ thru $U_{2,8}$ on card 4, etc. Format same as card 2.

APPENDIX D

Change 1
July 31, 1978